A Practical Guide To Software Licensing For Licensees And Licensors

With this practical book, architects, CTOs, and CIOs will learn a set of patterns for the practice of architecture, including analysis, documentation, and communication. Author Eben Hewitt shows you how to create holistic and thoughtful technology plans, communicate them clearly, lead people toward the vision, and become a great architect or Chief Architect. This book covers each key aspect of architecture comprehensively, including how to incorporate business architecture, information architecture, data architecture, application (software) architecture together to have the best chance for the system's success. Get a practical set of proven architecture practices focused on shipping great products using architecture Learn how architecture works effectively with development teams, management, and product management teams through the value chain Find updated special coverage on machine learning architecture Get usable templates to start incorporating into your teams immediately Incorporate business architecture, information architecture, data architecture, and application (software) architecture together

'I find your straightforward writing style an absolute joy, such a breath of fresh air!' - Angie Ash, PhD student '...thank you very much for your accessible language, clear lay out and practical applied approach. I suspect that this book will never be far from my side over the next 4 years!!' - Mayen Konarski, PhD student Using straight-forward language Doing Qualitative Research Using Your Computer walks readers through the process of managing and streamlining research

projects using commonly available Microsoft software applications. Drawing on a wide range of examples to demonstrate how easy it is to use such software, this guide is full of useful hints and tips on how to manage research more efficiently and effectively, including: - Formatting transcripts for maximum coding efficiency in Microsoft Word - Using features of Word to organize the analysis of data and to facilitate efficient qualitative coding - Synchronizing codes, categories, and important concepts between Microsoft Word and Microsoft Access - Efficiently storing and analyzing the qualitative data in Microsoft Excel - Creating flexible analytic memos in Access that help lead the researcher to final conclusions Ideal for those students or researchers who don't want to invest in expensive specialised software packages, this guide will be an invaluable companion for anyone embarking on their own research project.

David A. Sykes is a member of Wofford College's faculty. Incorporating new topics and original material, Introduction to Finite and Spectral Element Methods Using MATLAB, Second Edition enables readers to quickly understand the theoretical foundation and practical implementation of the finite element method and its companion spectral element method. Readers gain hands-on computational experience by using

A rich case-study analysis of open source software adoption by public organizations in different countries and settings. Government agencies and public organizations often consider adopting open source software (OSS) for reasons of transparency, cost, citizen access, and greater efficiency in communication and delivering services. Adopting Open Source Software offers five richly detailed real-world case studies of OSS adoption by public organizations. The authors analyze the cases and develop an overarching, conceptual framework to clarify the various enablers and inhibitors of

OSS adoption in the public sector. The book provides a useful resource for policymakers, practitioners, and academics. The five cases of OSS adoption include a hospital in Ireland; an IT consortium serving all the municipalities of the province of Bozen-Bolzano, Italy; schools and public offices in the Extremadura region of Spain; the Massachusetts state government's open standards policy in the United States; and the ICT department of the Italian Chamber of Deputies. The book provides a comparative analysis of these cases around the issues of motivation, strategies, technologies, economic and social aspects, and the implications for theory and practice.

A Practical Guide to TPM 2.0: Using the Trusted Platform Module in the New Age of Security is a straight-forward primer for developers. It shows security and TPM concepts, demonstrating their use in real applications that the reader can try out. Simply put, this book is designed to empower and excite the programming community to go out and do cool things with the TPM. The approach is to ramp the reader up quickly and keep their interest.A Practical Guide to TPM 2.0: Using the Trusted Platform Module in the New Age of Security explains security concepts, describes the TPM 2.0 architecture, and provides code and pseudo-code examples in parallel, from very simple concepts and code to highly complex concepts and pseudo-code. The book includes instructions for the available execution environments and real code examples to get readers up and talking to the TPM quickly. The authors then help the users expand on that with pseudo-code descriptions of useful applications using the TPM.

Although software localization is constantly changing, there are many issues, procedures and tools that will probably be applied for some time. This work provides an overview of the most common issues in today's software localization, from a

translator's, engineer's and project manager's view. Simplifying the often confusing array of software programs for fitting linear mixed models (LMMs), Linear Mixed Models: A Practical Guide Using Statistical Software provides a basic introduction to primary concepts, notation, software implementation, model interpretation, and visualization of clustered and longitudinal data. This easy-to-navigate reference details the use of procedures for fitting LMMs in five popular statistical software packages: SAS, SPSS, Stata, R/S-plus, and HLM. The authors introduce basic theoretical concepts, present a heuristic approach to fitting LMMs based on both general and hierarchical model specifications, develop the model-building process step-by-step, and demonstrate the estimation, testing, and interpretation of fixed-effect parameters and covariance parameters associated with random effects. These concepts are illustrated through examples using real-world data sets that enable comparisons of model fitting options and results across the software procedures. The book also gives an overview of important options and features available in each procedure. Making popular software procedures for fitting LMMs easy-to-use, this valuable resource shows how to perform LMM analyses and provides a clear explanation of mixed modeling techniques and theories. Build your own intelligent agent system... Intelligent agent technology is a tool of modern computer science that can be used to engineer complex computer programmes that behave rationally in dynamic and changing environments. Applications range from small programmes that intelligently search the Web buying and selling goods via electronic commerce, to autonomous space probes. This powerful technology is not widely used, however, as developing intelligent agent software

requires high levels of training and skill. The authors of this book have developed and tested a methodology and tools for developing intelligent agent systems. With this methodology (Prometheus) developers can start agent-oriented designs and implementations easily from scratch saving valuable time and resources. Developing Intelligent Agent Systems not only answers the questions "what are agents?" and "why are they useful?" but also the crucial question: "how do I design and build intelligent agent systems?" The book covers everything a practitioner needs to know to begin to effectively use this technology - including an introduction to the notion of agents, a description of the concepts involved, and a software engineering methodology. Read on for: a practical step-by-step introduction to designing and building intelligent agent systems. a full life-cycle methodology for developing intelligent agent systems covering specification, analysis, design and implementation of agents. PDT: Prometheus Design Tool – software support for the Prometheus design process. the example of an electronic bookstore to illustrate the design process throughout the book. Electronic resources including the Prometheus Design Tool (PDT), can be found at: http://www.cs.rmit.edu.au/agents/prometheus This book is aimed at industrial software developers, software engineers and at advanced undergraduate students. It assumes knowledge of basic software engineering but does not require knowledge of Artificial Intelligence or of mathematics. Familiarity with Java will help in reading the examples in chapter 10.

Highly recommended by JASA, Technometrics, and other journals, the first edition of this bestseller showed how to easily perform complex linear mixed model (LMM) analyses via a variety of software programs. Linear Mixed Models: A Practical Guide Using Statistical Software, Second Edition continues to lead readers step by step through the process of fitting LMMs. This second edition covers additional topics on the application of LMMs that are valuable for data analysts in all fields. It also updates the case studies using the latest versions of the software procedures and provides up-to-date information on the options and features of the software procedures available for fitting LMMs in SAS, SPSS, Stata, R/S-plus, and HLM. New to the Second Edition A new chapter on models with crossed random effects that uses a case study to illustrate software procedures capable of fitting these models Power analysis methods for longitudinal and clustered study designs, including software options for power analyses and suggested approaches to writing simulations Use of the lmer() function in the lme4 R package New sections on fitting LMMs to complex sample survey data and Bayesian approaches to making inferences based on LMMs Updated graphical procedures in the software packages Substantially revised index to enable more efficient reading and easier location of material on selected topics or software options More practical recommendations on using the software for analysis A new R package (WWGbook) that contains all of the data sets used in the examples Ideal for anyone who uses software for statistical modeling, this book eliminates the need to

read multiple software-specific texts by covering the most popular software programs for fitting LMMs in one handy guide. The authors illustrate the models and methods through real-world examples that enable comparisons of model-fitting options and results across the software procedures.

A Practical Guide to SysML: The Systems Modeling Language is a comprehensive guide to SysML for systems and software engineers. It provides an advanced and practical resource for modeling systems with SysML. The source describes the modeling language and offers information about employing SysML in transitioning an organization or project to model-based systems engineering. The book also presents various examples to help readers understand the OMG Systems Modeling Professional (OCSMP) Certification Program. The text is organized into four parts. The first part provides an overview of systems engineering. It explains the model-based approach by comparing it with the document-based approach and providing the modeling principles. The overview of SYsML is also discussed. The second part of the book covers a comprehensive description of the language. It discusses the main concepts of model organization, parametrics, blocks, use cases, interactions, requirements, allocations, and profiles. The third part presents examples that illustrate how SysML supports different model-based procedures. The last part discusses how to transition and deploy SysML into an organization or project. It explains the integration of SysML into a systems development environment. Furthermore, it describes the category of

data that are exchanged between a SysML tool and
other types of tools, and the types of exchange
mechanisms that can be used. It also covers the criteria
that must be considered when selecting a SysML.
Software and systems engineers, programmers, IT
practitioners, experts, and non-experts will find this book
useful. *The authoritative guide for understanding and
applying SysML *Authored by the foremost experts on
the language *Language description, examples, and
quick reference guide included
In the quest for quality, software developers have long
focused on improving the internal architecture of their
products. Larry L. Constantine--who originally created
structured design to effect such improvement--now joins
with well-known consultant Lucy A. D. Lockwood to turn
the focus of software development to the external
architecture. In this book, they present the models and
methods of a revolutionary approach to software that will
help programmers deliver more usable
software--software that will enable users to accomplish
their tasks with greater ease and efficiency. Recognizing
usability as the key to successful software, Constantine
and Lockwood provide concrete tools and techniques
that programmers can employ to meet that end. Much
more than just another set of rules for good user-
interface design, this book guides readers through a
systematic software development process. This process,
called usage-centered design, weaves together two
major threads in software development methods: use
cases (also used with UML) and essential modeling.
With numerous examples and case studies of both

conventional and specialized software applications, the authors illustrate what has been shown in practice to work and what has proved to be of greatest practical value. Highlights Presents a streamlined process for developing highly usable software Describes practical methods and models successfully implemented in industry Complements modern development practices, including the Unified Process and other object-oriented software engineering approaches

The purpose of large-scale software architecture is to capture and describe practical representations to make development teams more effective. In this book the authors show how to utilise software architecture as a tool to guide the development instead of capturing the architectural details after all the design decisions have been made. * Offers a concise description of UML usage for large-scale architecture * Discusses software architecture and design principles * Technology and vendor independent

For both students and engineers in R&D, this book explains machine vision in a concise, hands-on way, using the Vision Development Module of the LabView software by National Instruments. Following a short introduction to the basics of machine vision and the technical procedures of image acquisition, the book goes on to guide readers in the use of the various software functions of LabView's machine vision module. It covers typical machine vision tasks, including particle analysis, edge detection, pattern and shape matching, dimension measurements as well as optical character recognition, enabling readers to quickly and efficiently use these

functions for their own machine vision applications. A discussion of the concepts involved in programming the Vision Development Module rounds off the book, while example problems and exercises are included for training purposes as well as to further explain the concept of machine vision. With its step-by-step guide and clear structure, this is an essential reference for beginners and experienced researchers alike.

To fully leverage the value of software architecture in enterprise development projects, you need to expressly and consciously link architecture with project management. This book shows how, drawing on powerful lessons learned at Siemens, one of the world's leading software development organizations. The authors offer insight into project management for software architects, insight into software architecture for project managers, and above all, insight into integrating the two disciplines to maximize the effectiveness of both of them. Learn how to develop cost and schedule estimates for development projects, based on software architecture; how to clarify architecture so projects can be more effectively planned and managed; and then how to use architecture to organize, implement, and measure the project iteratively as work progresses.

CD-ROM contains: Canned HEAT v.2.0 -- Holodeck Lite v. 1.0.

Designing Software Architectures will teach you how to design any software architecture in a systematic, predictable, repeatable, and cost-effective way. This book introduces a practical methodology for architecture design that any professional software

engineer can use, provides structured methods supported by reusable chunks of design knowledge, and includes rich case studies that demonstrate how to use the methods. Using realistic examples, you'll master the powerful new version of the proven Attribute-Driven Design (ADD) 3.0 method and will learn how to use it to address key drivers, including quality attributes, such as modifiability, usability, and availability, along with functional requirements and architectural concerns. Drawing on their extensive experience, Humberto Cervantes and Rick Kazman guide you through crafting practical designs that support the full software life cycle, from requirements to maintenance and evolution. You'll learn how to successfully integrate design in your organizational context, and how to design systems that will be built with agile methods. Comprehensive coverage includes Understanding what architecture design involves, and where it fits in the full software development life cycle Mastering core design concepts, principles, and processes Understanding how to perform the steps of the ADD method Scaling design and analysis up or down, including design for pre-sale processes or lightweight architecture reviews Recognizing and optimizing critical relationships between analysis and design Utilizing proven, reusable design primitives and adapting them to specific problems and contexts Solving design problems in new domains, such as cloud,

mobile, or big data

The Really Useful ICT Book is a practical and easy-to-use guide to give you all the confidence you need to use ICT really effectively inside and outside the primary classroom. It makes clear how ICT can be taught as a standalone subject, and how it can be used easily and imaginatively to enhance teaching other subjects. Jam-packed with ideas and templates to save you time, this friendly handbook offers an introduction to: using ICT inside the classroom – including interactive whiteboards, computer suites, VLEs and e-safety using ICT outside the classroom – including word processors, laptops, data loggers and digital cameras when and how to use a wide range of software and hardware – from spreadsheet packages through to digital photography, e-portfolios and software simulation using ICT in all subject areas practical suggestions for using ICT in cross-curricular topics using ICT to develop teacher and pupil creativity using ICT for assessment and in your professional role. With an emphasis on developing children's creativity and on progression from Key Stage 1 to Key Stage 2, The Really Useful ICT Book is a comprehensive compendium of advice and inspiration for all training, newly qualified and experienced teachers, as well as those in support roles in primary schools.

Updated and expanded, this sixth edition is an essential resource for lawyers who work with clients

and conduct business transactions in this area. The guide includes new chapters on types of licenses, software development agreements, and software development kits and application program interfaces. New and updated model forms are available on an accompanying website that you can access allowing you to customize for your own practice needs. Software development consultant Wiegers describes various formal and informal methods for conducting a peer review program, such as pair programming, team reviews, the "walkthrough," and the ad hoc review. The main part of the text is devoted to the various stages of the technique of inspection. Coverage extends to the social issues involved in critiquing the work of others and overcoming resistance to reviews. c. Book News Inc. A Practical Guide to Plastics Sustainability: Concept, Solutions, and Implementation is a groundbreaking reference work offering a broad, detailed and highly practical vision of the complex concept of sustainability in plastics. The book's aim is to present a range of potential pathways towards more sustainable plastics parts and products, enabling the reader to further integrate the idea of sustainability into their design process. It begins by introducing the context and concept of sustainability, discussing perceptions, drivers of change, key factors, and environmental issues, before presenting a detailed outline of the current situation with types of plastics,

processing, and opportunities for improved sustainability. Subsequent chapters focus on the different possibilities for improved sustainability, offering a step-by-step technical approach to areas including design, properties, renewable plastics, and recycling and re-use. Each of these pillars are supported by data, examples, analysis and best practice guidance. Finally, the latest developments and future possibilities are considered. Approaches the idea of sustainability from numerous angles, offering practical solutions to improve sustainability in the development of plastic components and products Explains how sustainability can be applied across plastics design, materials selection, processing, and end of life, all set alongside socioeconomic factors Considers key areas of innovation, such as eco-design, novel opportunities for recycling or re-use, bio-based polymers and new technologies

Heather Meeker's Open Source for Business is a practical, readable guide to help businesspeople, engineers, and lawyers understand open source software licensing. Based on the author's twenty years as an attorney working at the crossroads of intellectual property and technology, this guide explains the legal and technical principles behind open source licensing so you can make the right decisions for your business. It offers tips on using open source, contributing to open source projects,

and releasing your own open source software. You'll also get access to quick-reference tables on the major open source licenses, plus forms and checklists you can use to promote compliance. In this book, you will learn . . . * Why open source is not a "virus" * What the GPL is and how to handle it * When and how to conduct open source audits * What a user-friendly open source policy looks like * How to avoid and respond to open source enforcement claims * How to use open source to fight patent infringement claims * How to manage trademarks for open source products

The cost of fixing software design flaws after the completion of a software product is so high that it is vital to come up with ways to detect software design flaws in the early stages of software development, for instance, during the software requirements, the analysis activity, or during software design, before coding starts. It is not uncommon that software requirements are ambiguous or contradict each other. Ambiguity is exacerbated by the fact that software requirements are typically written in a natural language, which is not tied to any formal semantics. A palliative to the ambiguity of software requirements is to restrict their syntax to boilerplates, textual templates with placeholders. However, as informal requirements do not enjoy any particular semantics, no essential properties about them (or about the system they attempt to describe) can be

proven easily. Formal methods are an alternative to address this problem. They offer a range of mathematical techniques and mathematical tools to validate software requirements in the early stages of software development. This book is a living proof of the use of formal methods to develop software. The particular formalisms that we use are EVENT B and refinement calculus. In short: (i) software requirements as written as User Stories; (ii) they are ported to formal specifications; (iii) they are refined as desired; (iv) they are implemented in the form of a prototype; and finally (v) they are tested for inconsistencies. If some unit-test fails, then informal as well as formal specifications of the software system are revisited and evolved. This book presents a case study of software development of a chat system with EVENT B and a case study of formal proof of properties of a social network. The amount of software used in safety-critical systems is increasing at a rapid rate. At the same time, software technology is changing, projects are pressed to develop software faster and more cheaply, and the software is being used in more critical ways. Developing Safety-Critical Software: A Practical Guide for Aviation Software and DO-178C Compliance equips you with the information you need to effectively and efficiently develop safety-critical, life-critical, and mission-critical software for aviation. The principles also apply to software for

automotive, medical, nuclear, and other safety-critical domains. An international authority on safety-critical software, the author helped write DO-178C and the U.S. Federal Aviation Administration's policy and guidance on safety-critical software. In this book, she draws on more than 20 years of experience as a certification authority, an avionics manufacturer, an aircraft integrator, and a software developer to present best practices, real-world examples, and concrete recommendations. The book includes: An overview of how software fits into the systems and safety processes Detailed examination of DO-178C and how to effectively apply the guidance Insight into the DO-178C-related documents on tool qualification (DO-330), model-based development (DO-331), object-oriented technology (DO-332), and formal methods (DO-333) Practical tips for the successful development of safety-critical software and certification Insightful coverage of some of the more challenging topics in safety-critical software development and verification, including real-time operating systems, partitioning, configuration data, software reuse, previously developed software, reverse engineering, and outsourcing and offshoring An invaluable reference for systems and software managers, developers, and quality assurance personnel, this book provides a wealth of information to help you develop, manage, and approve safety-critical software more

confidently.

The non-technical guide to building a booming tech-enabled business Thinking of starting a technology-enabled business? Or maybe you just want to increase your technology mojo so you can do your job better? You do not need to learn programming to participate in the development of today's hottest technologies. But there are a few easy-to-grasp foundation concepts that will help you engage with a technical team. Starting a Tech Business explains in practical, actionable terms how to formulate and reality test new ideas package what you learn into frameworks that are highly actionable for engineers understand key foundation concepts about modern software and systems participate in an agile/lean development team as the 'voice of the customer' Even if you have a desire to learn to program (and I highly recommend doing whatever unlocks your 'inner tinkerer'), these foundation concepts will help you target what exactly you want to understand about hands-on technology development. While a decade ago the barriers to creating a technology-enabled business required a pole vault, getting started today only requires a determined step in the right direction. Starting a Tech Business supplies the tools prospective entrepreneurs and business enterprises need to avoid common pitfalls and succeed in the fast-paced world of high-tech business. Successful execution requires thoughtful,

evidence-based product formulation, well-articulated design, economic use of systems, adaptive management of technical resources, and empathetic deployment to customers. Starting a Tech Business offers practical checklists and frameworks that business owners, entrepreneurs, and professionals can apply to any tech-based business idea, whether you're developing software and products or beginning a technology-enabled business. You'll learn: 1. How to apply today's leading management frameworks to a tech business 2. How to package your product idea in a way that's highly actionable for your technical team 3. How to ask the right questions about technology selection and product architecture 4. Strategies to leverage what your technology ecosystem has to offer 5. How to carefully define the roles on your team, and then effectively evaluate candidates 6. The most common disconnects between engineers and business people and how to avoid them 7. How you can apply process design to your tech business without stifling creativity 8. The steps to avoid the most common pitfalls tech founders encounter Now is one of the best times to start a technology-enabled business, and anyone can do it with the right amount and kind of preparation. Starting a Tech Business shows you how to move a product idea to market quickly and inexpensively—and to tap into the stream of wealth that a tech business can provide.

Translation technology has evolved quickly with a large number of translation tools available. In this revised addition, much content has been added about translating and engineering HTML and XML documents, multilingual web sites, and HTML-based online help systems. Other major changes include the addition of chapters on internationalizatoi, software quailty assurance, descktop publishing and localization supprort. There is a focus on translators who want to learn about localization ad translation technology.

It is estimated that one in five people has a disability; if you do the math, that's a huge, addressable market that is often overlooked. Digital Accessibility practitioners tap into this market by making digital documents-as well as web and mobile apps-accessible to everyone. However, many enterprises struggle to create, maintain, and scale their digital accessibility efforts. Agile Accessibility Handbook outlines the steps organizations can take to capture this market, avoid risk, maintain agility, and close the accessibility gap. Using the information provided within this handbook, accessibility subject matter experts, development team members, and executives in charge of setting priorities can together learn how to build successful accessibility experiences for everyone.

This is a comprehensive guide to Scrum for all (team members, managers, and executives). If you want to

use Scrum to develop innovative products and services that delight your customers, this is the complete, single-source reference you've been searching for. This book provides a common understanding of Scrum, a shared vocabulary that can be used in applying it, and practical knowledge for deriving maximum value from it.

Ship It! is a collection of tips that show the tools andtechniques a successful project team has to use, and how to use themwell. You'll get quick, easy-to-follow advice on modernpractices: which to use, and when they should be applied. This bookavoids current fashion trends and marketing hype; instead, readersfind page after page of solid advice, all tried and tested in thereal world. Aimed at beginning to intermediate programmers, Ship It! will show you: Which tools help, and which don't How to keep a project moving Approaches to scheduling that work How to build developers as well as product What's normal on a project, and what's not How to manage managers, end-users and sponsors Danger signs and how to fix them Few of the ideas presented here are controversial or extreme; most experiencedprogrammers will agree that this stuff works. Yet 50 to 70 percent of allproject teams in the U.S. aren't able to use even these simple, well-acceptedpractices effectively. This book will help you get started. Ship It! begins by introducing the common technicalinfrastructure that every project

needs to get the job done. Readerscan choose from a variety of recommended technologies according totheir skills and budgets. The next sections outline the necessarysteps to get software out the door reliably, using well-accepted,easy-to-adopt, best-of-breed practices that really work. Finally, and most importantly, Ship It! presents commonproblems that teams face, then offers real-world advice on how tosolve them.

The Laboratory Computer: A Practical Guide for Physiologists and Neuroscientists introduces the reader to both the basic principles and the actual practice of recording physiological signals using the computer. It describes the basic operation of the computer, the types of transducers used to measure physical quantities such as temperature and pressure, how these signals are amplified and converted into digital form, and the mathematical analysis techniques that can then be applied. It is aimed at the physiologist or neuroscientist using modern computer data acquisition systems in the laboratory, providing both an understanding of how such systems work and a guide to their purchase and implementation. The key facts and concepts that are vital for the effective use of computer data acquisition systems A unique overview of the commonly available laboratory hardware and software, including both commercial and free software A practical guide to designing one's own or

choosing commercial data acquisition hardware and software

Heather Meeker's Open Source for Business is a practical, readable guide to help businesspeople, engineers, and lawyers understand open source software licensing. Based on the author's twenty years as an attorney working at the crossroads of intellectual property and technology, this guide explains the legal and technical principles behind open source licensing so you can make the right decisions for your business. It offers tips on using open source, contributing to open source projects, and releasing your own open source software. You'll also get access to quick-reference tables on the major open source licenses, plus forms and checklists you can use to promote compliance. In this book, you will learn . . . - Why open source is not a "virus" - What the GPL is and how to handle it - When and how to conduct open source audits - What a user-friendly open source policy looks like - How to avoid and respond to open source enforcement claims - How to use open source to fight patent infringement claims - How to manage trademarks for open source products

The software industry has undergone rapid development since the beginning of the twenty-first century. These changes have had a profound impact on translators who, due to the evolving nature of digital content, are under increasing pressure to

adapt their ways of working. Localizing Apps looks at these challenges by focusing on the localization of software applications, or apps. In each of the five core chapters, Johann Roturier examines: The role of translation and other linguistic activities in adapting software to the needs of different cultures (localization); The procedures required to prepare source content before it gets localized (internationalization); The measures taken by software companies to guarantee the quality and success of a localized app. With practical tasks, suggestions for further reading and concise chapter summaries, Localizing Apps takes a comprehensive look at the transformation processes and tools used by the software industry today. This text is essential reading for students, researchers and translators working in the area of translation and creative digital media.

Technology Transactions also provides a complete discussion of the many privacy considerations that must be kept in mind in an agreement to leverage any emerging technology. Considerations under the following statutes are discussed: - HIPAA- The Gramm-Leach-Bliley Act- The Childrens Online Privacy and Protection Act (COPPA) as well as the many protections that are afforded to international data transfers

Citation Management Tools: A Practical Guide for Librarians will provide librarians with the essential skills and information required to support the use of citation managers at their libraries, and to explore creative ways to use these tools. Rev. ed of: A practical guide to software licensing for

licensees and licensors / H. Ward Classen. 3rd ed. c2008. Written by a leading expert in the field, this unique volume contains current test design approaches and focuses only on software test design. Copeland illustrates each test design through detailed examples and step-by-step instructions. This guidebook examines the fundamental issues that both licensors and licensees confront in the negotiation of a software license and, where appropriate, relevant ancillary issues such as software development as well as maintenance and support. A companion CD-ROM is included with customizable agreements and relevant forms.
This title provides systematic performance planning techniques for diverse computing environments and architectures. It seeks to smoothly integrate performance analysis into an existing software development process.
Copyright: 398d984033206dc1d08eaa6535032ce1