

## Entwurfsmuster Elemente Wiederverwendbarer Software Gamma

*Some analysts predict that electronic commerce will grow to more than \$300 billion in the next five years. With electronic commerce growing in exponentially, staying competitive through an effect e-business strategy is a tough challenge. Electronic commerce represents one of the most promising directions for generating competitive advantage at the micro level of the organization and for increasing productivity at the macro level of the economy. Electronic Commerce: Opportunity and Challenges looks at the critical fundamental elements that could impact a user's perception of business-to-business and business-to-consumer electronic commerce. It presents a study of key components and concepts of e-commerce, evaluating the critical success factors for global e-commerce, the economics of e-commerce and the practical issues involved with e-commerce in various applications.*

*Gegenstand dieses Buches ist das Management heterogener betrieblicher Applikationslandschaften. Hierbei beschränkt sich der Text nicht auf die Integrationsaspekte, die unter dem Stichwort Enterprise Application Integration (EAI) diskutiert werden, sondern er geht darüber hinaus auf Fragen der Planung, Bewertung und Steuerung von Applikationslandschaften ein. Es werden unterschiedliche Methoden und Techniken für die verschiedenen Aspekte der Applikationsintegration behandelt. Zugleich werden die Ergebnisse des Kompetenzzentrums "Application Integration Management" zusammengefasst, in dem vom Institut für Wirtschaftsinformatik der Universität St. Gallen gemeinsam mit fünf bedeutenden europäischen Unternehmen (Credit Suisse, Informatikzentrum der deutschen Sparkassen, Swisscom IT Services, Swiss Re, Winterthur Versicherungen) in den Jahren 2002 bis 2004 Konzepte, Methoden und Techniken für die Applikationsintegration in großen Dienstleistungsunternehmen erarbeitet wurden. Besonderes Gewicht wird dabei auf die praktische Umsetzbarkeit der Ergebnisse gelegt. So werden Beiträge von Forschenden des Instituts für Wirtschaftsinformatik durch Beiträge von Praktikern aus den beteiligten Unternehmen ergänzt.*

*Das Buch Auf dem Weg zur Integration Factory fasst den State of the Art sowie die Zukunftsperspektiven im Bereich der integrierten Informationslogistik zusammen. Die Autoren untersuchen zum einen, inwieweit sich bisherige Ansätze zum Data Warehousing sowie zur Enterprise Application Integration mittelfristig technisch, organisatorisch und wirtschaftlich als geeignete Lösungen erwiesen haben. Die Kernthemen sind hierbei Architekturen, Vorgehensmodelle, Business Intelligence, Online-Analyse, Customer Relationship Management, Supply Chain Management und Anwendungsintegration. Zum anderen werden neuere Ansätze vorgestellt, welche die Integration des Data Warehouse in die Gesamt-Informationslogistik zum Ziel haben und so die Realisierung neuer Applikationstypen und Geschäftsmodelle ermöglichen. Darüber hinaus wird insbesondere auf eine Verbesserung der Prozessunterstützung durch einen höheren Integrationsgrad der Applikationen eingegangen. Das Buch wendet sich vorwiegend an Praktiker aus den Bereichen Betriebswirtschaft und Wirtschaftsinformatik.*

Elemente wiederverwendbarer objektorientierter Software

Pattern-orientierte Software-Architektur

Kubernetes Patterns

Design Patterns

Visionäre der Programmierung - Die Sprachen und ihre Schöpfer

Elements of Reusable Object-Oriented Software

Auf dem Weg zur Integration Factory

Unterricht und seine Gestaltung stehen in einem Spannungsverhältnis zwischen Theorie und Praxis. Wie kann diese Kluft zwischen Wissen und Handeln überbrückt werden? Als eine Möglichkeit werden in jüngster Zeit didaktische Entwurfsmuster diskutiert. Auf systematische Weise dokumentieren sie bewährte Lösungen für wiederkehrende Probleme bei der Gestaltung didaktischer Szenarien und machen so das implizite Wissen erfahrener Lehrender für Novizinnen und Novizen nutzbar. Von wem, wie und warum wurde der Muster-Ansatz aus der Architektur auf die Didaktik übertragen? Welche Problematik verbirgt sich in diesem gedanklichen Brückenschlag? Wie sehen didaktische Entwurfsmuster aus und worin bestehen ihre Stärken und Schwächen? Was sind die notwendigen Voraussetzungen dafür, dass didaktische Entwurfsmuster nutzbringend für die Unterrichtsgestaltung eingesetzt werden können? Diesen Fragen nähert sich Reinhard Bauer aus einer diskursanalytischen Perspektive mit dem Ziel, den von Christopher Alexander für die Architektur entwickelten Muster-Ansatz für die Didaktik zu öffnen und nutzbar zu machen. Reinhard Bauer promovierte 2014 an der Alpen-Adria-Universität Klagenfurt. Er lehrt und forscht am Institut für übergreifende Bildungsschwerpunkte (IBS) der Pädagogischen Hochschule Wien, wo er sich im Zentrum für Lerntheologie und Innovation (ZLI) mit den Einsatzmöglichkeiten von digitalen Technologien, Medien und Werkzeugen für eine zeitgemäße und innovative (Hochschul-)Didaktik beschäftigt.

Nick Gehrke erarbeitet Konzepte, mit denen Peer-to-Peer-Technologien nutzbar gemacht werden können, um elektronische Marktplätze hochgradig verteilt aufzubauen, so dass möglichst weitgehend auf zentrale Intermediäre verzichtet werden kann. Im Mittelpunkt stehen kommerzielles Filesharing, Peer-to-Peer-Marktplätze und Grid Computing.

In Visionäre der Programmierung - Die Sprachen und ihre Schöpfer werden exklusive Interviews mit den Entwicklern von historischen wie auch von hochaktuellen Programmiersprachen veröffentlicht. In dieser einzigartigen Zusammenstellung erfahren Sie die Hintergründe, die zu den spezifischen Design-Entscheidungen in den Programmiersprachen geführt haben und über die ursprüngliche Ziele, die die Entwickler im Kopf hatten, als sie eine neue Programmiersprache entwarfen. Ebenso können Sie lesen, wieso Abweichungen zum ursprünglichen Design entstanden und welchen Einfluss die jeweilige Sprache auf die heutige Softwareentwicklung noch besitzt. Adin D. Falkoff: APL Thomas E. Kurtz: BASIC Charles H. Moore: FORTH Robin Milner: ML Donald D. Chamberlin: SQL Alfred Aho, Peter Weinberger und Brian Kernighan: AWK Charles Geschke und John Warnock: PostScript Bjarne Stroustrup: C++ Bertrand Meyer: Eiffel Brad Cox und Tom Love: Objective-C Larry Wall: Perl Simon Peyton Jones, Paul Hudak, Philip Wadler und John Hughes: Haskell Guido van Rossum: Python Luiz Henrique de Figueiredo und Roberto Ierusalimsky: Lua James Gosling: Java Grady Booch, Ivan Jacobson und James Rumbaugh: UML Anders Hejlsberg: Delphi-Entwickler und führender Entwickler von C#

Opportunity and Challenges

Prinzipien des Softwaredesigns

Agile Entwicklungspraktiken mit Scrum

Wege zum Text

An Introduction to Object-oriented Analysis and Design and Iterative Development

Distributed Manufacturing

Entwurfsmuster als Elemente wiederverwendbarer objektorientierter Software

Design Patterns bieten Lösungsmuster für häufig auftretende Softwareaufgaben in der Software-Entwicklung, aber sie gelten als schwer verständlich und als sehr theoretisch. Für die 2. Auflage seines Bestsellers hat Stephan Schmidt alle Neuerungen der PHP-Version 5.3 berücksichtigt und außerdem den Design-Patterns-Katalog noch einmal deutlich erweitert. Er erläutert zunächst einige allgemeine Regeln des guten Software-Designs, um dann alle wichtigen Design Patterns detailliert vorzustellen und anhand von Praxisbeispielen Schritt für Schritt und gut nachvollziehbar zu implementieren. Da PHP häufig für die Entwicklung von Webanwendungen eingesetzt wird, liegt ein Schwerpunkt des E-Books auf Design Patterns in diesem Umfeld: Der Autor zeigt, wie Websites anhand des Schichtenmodells sinnvoll aufgebaut werden und in welcher Form hier Enterprise-Patterns zum Einsatz kommen.

Neben parallel arbeitende Software kann die Leistung heutiger Multicores-Prozessoren ausnuten. Das Buch vermittelt ein solides Grundwissen über Softwareentwicklung im Multicores-Zeitalter und dient als Nachschlagewerk für die tägliche Arbeit. Der erste Teil führt gut lesbar durch alle praxisrelevanten Grundlagen der Programmierung und der Architektur paralleler Software. Mit dem zweiten Teil erhält der Leser einen kompakt und strukturiert aufbereiteten Leitfaden für die produktive Entwicklung in den Sprachen C/C++, Java und C#.

Dieses Buch führt Sie in die wichtigsten Entwurfsprinzipien und Konzepte der Softwaretechnik ein Entwurfsprinzipien, alle Design Patterns, sind bewährte, einfache und klare Denkkonzepte des Software Engineering, welche Entwicklern helfen, hochwertige Softwaresysteme zu konstruieren. Sie greifen in die Konstruktion eines Systems ein und betreffen die für den Entwickler sichtbare Qualität des Quellcodes. Qualitätsziele für den Entwurf sind beispielsweise:
· das Vermeiden unerwünschter Kopplungen von Softwareteilen
· die Einfachheit und Verständlichkeit
· die Testbarkeit
· die Stabilität bei Programmweiterungen
Dieses Buch führt Sie in die Grundlagen der Softwaretechnik ein und stellt Ihnen die wichtigsten Entwurfsprinzipien und Konstruktionskonzepte vor. Es gibt keine allgemein anerkannten Kataloge von Entwurfsprinzipien für die Softwareentwicklung. Daher enthält dieses Buch eine Auswahl von praxisrelevanten Entwurfsprinzipien und Konzepten, die zudem in der Clean-Code-Bewegung eine große Rolle spielen. Das Buch eignet sich nicht nur für die tägliche Arbeit in Bereich der Softwaretechnik, sondern auch als Lehrbuch für Studierende der Informatik. Die folgenden Inhalte behandelt der Autor in seinem Buch beschäftigt sich Joachim Goll unter anderem mit den folgenden Themeniken:
· Das Konzept der Software-Module
· Unerwünschte Abhängigkeiten und deren Abschwächung in Programmen
· Erzeugung einer modularen Struktur in Programmen
· Reduktion der Komplexität von Programmen
· Konstruktion schwach gekoppelter Teilsysteme
· Korrektheit von polymorphen Programmen
· Stabilität von Programmen und Erweiterbarkeit bei Programmänderungen
· Inversion of Control

Fieldbus Systems and Their Applications 2001 (PeT'2001)

Proceedings der DM2004 – Data Warehousing und EAI

Applying UML and Patterns

Electronic Commerce and Web Technologies

Mit lauffähigen Beispielen in Java

Evaluierung von Softwarearchitekturen

C# 3.0 Entwurfsmuster

A catalog of solutions to commonly occurring design problems, presenting 23 patterns that allow designers to create flexible and reusable designs for object-oriented software. Describes the circumstances in which each pattern is applicable, and discusses the consequences and trade-offs of using the pattern within a larger design. Patterns are compiled from real systems, and include code for implementation in object-oriented programming languages like C++ and Smalltalk. Includes a bibliography. Annotation copyright by Book News, Inc., Portland, OR

This volume contains the papers from the 4th IFAC conference on Fieldbus Systems and Their Applications (FeT'2001) held in Nancy, France, 15 – 16 November 2001. This conference was, for the first time, sponsored by IFAC with previous conferences being held in Vienna (Austria) 1995, 1997 and Magdeburg (Germany) 1999. The conference was supported by IFAC, the technical committees on Components and Instruments, on Advanced Manufacturing Technology, on Real-Time Software Engineering and their chairmen. The program covered a variety of research topics, which are of current interest, such as:
·performances,
·dependability of fieldbus based systems,
·definition of devices profiles and architectures problems,
·conformance testing and interoperability,
·scheduling,
·new technologies and wireless systems. More and more papers related to the validation and to the evaluation of application, using formal methods and techniques. These papers came from academic and industrial authors from many countries and the diversity of the points of view expressed leads to an interesting and lively proceedings. Four plenary papers, written by well-known specialists, provide tutorial material, as well as some history and background, and introduce the more specialised content which follows. Altogether nearly 50 papers are presented.

Der Bestseller von Gamma und Co. in komplett neuer ÜbersetzungDer Klassiker für alle ProgrammiererDas Standardwerk für die objektorientierte Softwareentwicklung Mit Design Patterns lassen sich wiederkehrende Aufgaben in der objektorientierten Softwareentwicklung effektiv lösen. In diesem Buch stellen die Autoren einen Katalog einfacher und prägnanter Lösungen für häufig auftretende Problemstellungen vor. Diese 23 Patterns ermöglichen flexiblere, elegantere und wiederverwendbare Designs, so dass Problemlösungen nicht jedes Mal aufs Neue entwickelt werden müssen. Bei jedem Pattern ist angegeben, in welchem Kontext es besonders geeignet ist und sinnvoll eingesetzt werden kann.

Mehr effektiv C++ programmieren

Strategien für schwach gekoppelte, korrekte und stabile Software

Design Patterns in der objektorientierten Softwaretechnik

Überlegungen zur Verfügbarkeit mediävistischer Editionen im 21. Jahrhundert. Grazer Kolloquium 17.-19. September 2008

Entwurfstrategien für komplexe Systeme

Objektorientierte Programmierung in Oberon-2

Entwurfsmuster

*Das Buch vermittelt eine neue Sicht zur Software-Technik, in dem es vor allem den Engineering Aspekt stärker berücksichtigt. Das Buch ist Web-bezogen aufgebaut und ermöglicht damit, das Internet für den Wissenerwerb und ausgewählte Übungen zu nutzen.*

*We welcome you to the Second International Conference on E commerce and Web Technology (ECWEB 2001) held in conjunction with DEXA 2001 in Munich, Germany. This conference, now in its second year, is a forum to bring together researchers from academia and commercial developers from industry to discuss the state of the art in E commerce and web technology and explore new ideas. We thank you all for coming to Munich to participate and debate the new emerging advances in this area. The research presentation and discussion during the conference will help to exchange new ideas among the researchers, developers, and practitioners. The conference program consists of an invited talk by Hannes Werthner, University of Trento, Italy, as well as the technical sessions. The regular sessions cover topics from XML Transformations and Web Development to User Behavior and Case Studies. The workshop has attracted more than 80 papers and each paper has been reviewed by at least 3 program committee members for its merit. The program committee have selected 31 papers for presentation. We would like to express our thanks to the people who helped put together the technical program: the program committee members and external reviewers for their timely and rigorous reviews, the DEXA organizing committee for their help in administrative work and support, and special thanks to Gabriela Wagner for always responding promptly.*

*Inhaltsangabe-Abstract: In 1994 the Gang of Four, consisting of Erich Gamma, Richard Helm, Ralph Johnson and John Vlissides, published the book Design Pattern – Elements of Reusable Object-Oriented Software. Within that book the four information scientists described 23 design patterns, which they classifed into the categories Creational Design Pattern, Structural Design Pattern and Behavioral Design Pattern. Even though design patterns exist since 15 years at present, they have not lost relevance. Due to new concepts the usage of design patterns within web application is increasing. Meanwhile all 23 established design patterns are available as PHP implementations. Aside web languages like AJAX, JavaScript, XHTML and CSS also appreciate the strength of design patterns. At first web languages like AJAX, PHP, et cetera will be analyzed if they qualify for the usage of design patterns. At second the usage of design patterns within open source web applications like Typo3, Joomla, Wordpress, et cetera will be examined. At third various web developers of 161, web.de, Telekom, et cetera will be interviewed to investigate if and in which amount design patterns are used by companies. Intention of this thesis is to determine the relevance of design pattern within web applications. Thereby advantages will also be shown like disadvantages. Also the question, if design patterns should be used by default or only if the concerning project reached a specific size, will be answered.
Inhaltsverzeichnis:Table of Contents: Abstract Preface1.1.Introduction1.2.Analysis of web programming languages2
2.1.Definition of recognition characteristics3
2.1.1.Regular classes4
2.1.2.Abstract classes5
2.1.3.Static attributes and methods6
2.1.4.Scopes7
2.1.5.Inheritance through expansion8
2.1.6.Inheritance through implementation9
2.2.Examination of de7ned characteristics10
2.2.1.ActionScript11
2.2.2.AJAX/JavaScript13
2.2.3.ASP15
2.2.4.PHP17
2.2.5.Ruby19
3.Analysis of open source applications21
3.1.Definition of recognition characteristics22
3.1.1.Analysis by hand23
3.1.2.Analysis by recognition characteristics24
3.1.3.Analysis by UML to code generator26
3.1.4.Analysis by manuals27
3.2.Examination of de7ned characteristics28
3.2.1.Coppertine Photo Gallery29
3.2.2.Drupal31
3.2.3.phpBB33
3.2.4.WordPress34
3.2.5Zend Framework35
4.Examples of design patterns37
4.1.Model View Controller Pattern38
4.2.Intercepting Filter Pattern40
4.3.Registry Pattern42
4.4.Template View [...]*

Multicores-Software

Cases, Technologies, and Opportunities

Peer-to-Peer-Applikationen für elektronische Märkte

A Proceedings Volume from the 4th IFAC Conference, Nancy, France, 15-16 November 2001

Entwurfsmuster von Kopf bis Fuß

Grundlagen

Relevance of design patterns within web applications

Die besten Muster und Prinzipien in einem Buch Umfangreicher Überblick über die in der Praxis relevanten Patterns mit Beispielen aus dem realen Arbeitsalltag inkl. aller Code-Beispiele zum kostenlosen Download Source Code und Issue Tracker auf GitHub In diesem praktischen Handbuch stellen Ihnen die Autoren Bilgin Ibraym und Roland Huß von Red Hat verbreitete, wiederverwendbare Muster und Prinzipien zum Design und zur Implementierung von Cloud-nativen Anwendungen auf Kubernetes vor. Zu jedem Pattern gibt es eine Beschreibung des Problems und eine Lösung speziell für Kubernetes. Alle Patterns sind durch konkrete Code-Beispiele ergänzt. Dieses Buch ist ideal für Entwickler und Architekten, die schon mit den grundlegenden Konzepten von Kubernetes vertraut sind und nun lernen wollen, wie sie häufig auftretende Cloud-native Probleme mit bewährten Entwurfsmustern angehen können. Über diese Pattern-Kategorien werden Sie etwas erfahren: Grundlegende Patterns behandeln die zentralen Prinzipien und Praktiken für das Bauen Container-basierter Cloud-nativer Anwendungen. Verhaltens-Patterns untersuchen detaillierte Konzepte für das Managen von Container- und Plattform-Interaktionen. Strukturelle Patterns helfen Ihnen dabei, Container in einem Pod so zu organisieren, dass spezifische Anwendungsfälle gelöst werden können. Konfigurations-Patterns liefern Einblicke in den Umgang mit Anwendungskonfigurationen in Kubernetes. Fortgeschrittene Patterns behandeln komplexere Themen wie Operatoren oder Autoscaling.

The Internet is revolutionizing retail merchandising and shopping. Software agents are capable of automating the more routine, tedious and time-consuming tasks involved in the trading process. Internet Commerce and Software Agents: Cases, Technologies and Opportunities addresses some major Internet commerce issues and the challenges to be met in achieving automated and secure Internet trading.

Changing world market conditions have forced manufacturers to apply new architectures and technologies for the design and control of manufacturing systems. Distributed Manufacturing: Paradigm, Concepts, Solutions and Examples outlines the current requirements of manufacturing systems and addresses the architectures, methodologies, and technologies developed within European research activities in response to these requirements. Distributed Manufacturing: Paradigm, Concepts, Solutions and Examples will be of interest to researchers and developers in all fields involving industrial control systems, as well as to decision-makers within industry and government organizations. The reader will gain a detailed knowledge of the current research directions in industrial control, reaching a comprehensive understanding of current advances, their expected benefits and limitations, and the possible consequences for industrial businesses.

Der Muster-Ansatz von Christopher Alexander und Implikationen für die Unterrichtsgestaltung

Einführung und Referenz

PHP Design Patterns

Internet Commerce and Software Agents
Am Beispiel einer Schach-Community

Planung, Bewertung und Steuerung von Applikationslandschaften

Scrum ist ein agiles Management-Framework, das keine Entwicklungspraktiken empfiehlt oder gar vorschreibt. Auswahl und Einsatz der richtigen Praktiken fallen unter die Selbstorganisation des Teams. Ohne den Einsatz geeigneter Entwicklungspraktiken und -tools ist der Einsatz von Scrum in der Softwareentwicklung jedoch nicht dauerhaft erfolgreich. Dieses Buch beschreibt praxismah die wichtigsten Praktiken wie Architekturvison, inkrementeller Entwurf, Continuous Integration, testgetriebene Entwicklun, Refactoring, Akzeptanztests sowie modellgetriebene und verteilte Entwicklung mit Scrum.

Nach erster Euphorie und der folgenden Ernüchterung herrscht im Augenblick wieder starkes Interesse an objektorientierten Themen. Der Schwerpunkt der Diskussionen und Entwicklungen liegt jetzt auf Methoden. Vereinfacht wird diese Diskussion durch die absehbare Standardisierung einer Notation zur Darstellung objektorientierter Modelle, der Unified Modeling Language (UML). Erschwert wird die Diskusion und die sachgerechte Entscheidung über den Einsatz objektorientierter Methoden durch den unterschiedlichen Hintergrund der Beteiligten. Hinzu kommt eine vielfältig verbundene Begriffs welt, die verwirrend erscheinen kann und es wohl zum Teil auch ist. Hier setzt dieses Buch an. Es ist aus Vorlesungen über Software-Engineer ing an der Fachhochschule Harnburg hervorgegangen. Die Studierenden tru gen durch ihr Konsequentes Nachfragen nach der Bedeutung der eingeführten oder einfach so verwendeten Begriffe dazu bei, dass ich mich systematisch um Definitionen und Hinweise auf abweichende Verwendung aller Begriffe bemüht habe. So ist ein Buch entstanden, das sich an Leser wendet, die aus unterschied lichen Gründen, aber ohne umfangreiche theoretische Vorbildung, objektori entierte Entwicklung lernen wollen oder müssen:
• Studierende an Universitäten oder Fachhochschulen,
• Software-Entwickler, die Programmiersprachen und klassische Methoden, wie Entry-Relationship-Modellierung oder strukturierte Analyse, einsetz nen und Objektorientierung lernen wollen,
• Entwickler, die objektorientierte Programmierung kennen und nun die UML kennen lernen wollen. Dabei habe ich vermutet, mir wichtig erscheinende Themen mit einigen Akzen ten zu behandeln, die ich in anderen Werken vermisst habe. Die wichtigsten Punkte, die behandelt werden, sind:
• Darstellung eines systematischen, qualitätsorientierten Ansatzes.
• Klare Definitionen aller vorkommenden Begriffe und Konzepte.

In this collected volume, editors are joined by expert archivists, librarians, publishers and IT specialists to discuss the problem of making editions of medieval texts available both in print and digitally. To facilitate orientation, the discussion included consideration of setting up an internet-based 'Text Portal', which would include access to the 'raw materials' of editorship in the form of historical text media or their facsimiles.

Didaktische Entwurfsmuster

Hybride Simulation mobiler Geschäftsprozesse

Electronic Commerce: Opportunity and Challenges

Second International Conference, EC-Web 2001 Munich, Germany, September 4-6, 2001 Proceedings

Perspektiven für eine hochgradig dezentralisierte digitale Wirtschaft

Paradigm, Concepts, Solutions and Examples

Architektur- und Entwurfsmuster der Softwaretechnik

**Zum Lernen, Nachschlagen und die erfolgreiche Praxis des Software Engineering. Das Buch ist so aufbereitet, dass es die wesentlichen Teilgebiete des internationalen "Software Engineering Body of Knowledge" (SWEBOK) abdeckt: die Grundlage für eine Ausbildung im Software Engineering nach internationalem Standard. Hier erfahren Sie alles über die Grundprinzipien, Methoden und Technologien jeweils im Kontext ihrer erfolgreichen Umsetzung und Anwendung. Die Darstellung folgt der UML-Methode mit den jeweiligen Tool-Anwendungen. Die neue Auflage wurde gänzlich überarbeitet und aktualisiert.**

**Sie haben natürlich keine Lust, das Rad (oder gar einen Platten) neu zu erfinden, klar. Warum sich also nicht mal bei Entwurfsmustern umsehen? Entwurfsmuster sind die geballte Erfahrung von Leuten, die sich vor Ihnen denselben Schwierigkeiten beim Software-Design gegenüber sahen – und dafür elegante, übertragbare Lösungen fanden. Mit Entwurfsmustern können Sie viel Zeit sparen ... Zeit für spannendere, komplexere oder einfach unterhaltsamere Herausforderungen. Dazu brauchen Sie folgende Informationen: Welche Entwurfsmuster sind wichtig und nützlich? Wann und warum verwende ich diese Muster? W.**

**Mit durchdachtem Softwaredesign zu besserem Code US-Bestseller des renomierten Stanford-Professors John Ousterhout kombiniert eingängige Codebeispiele (v.a. Java & C++) und High-Level-Konzepte, durch die Sie Ihr Verständnis von Softwaredesign deutlich vertiefen werden Bewährte Didaktik: Die Inhalte wurden im Rahmen eines Kurses an der Stanford University mehrfach mit Studierenden durchgespielt. John Ousterhout, Erfinder von Tcl und seit 2008 Professor am Department of Computer Science der Stanford University, sieht Softwaredesign als zentrale Kompetenz, die großartige von normalen Programmierern und Programmiererrinnen unterscheidet – eine Kompetenz, die aber kaum gelehrt und vermittelt wird. In seinem Buch gibt er persönliche Erfahrungen weiter, die er durch das Schreiben und Lesen von mehr als 250.000 Zeilen Code gewonnen hat. Seine nahezu philosophischen Überlegungen, bewährten Prinzipien und eingängigen Codebeispiele sind Ergebnis eines Kurses, den er an der Stanford University mehrfach gehalten hat.**

Refactoring to patterns

Wiederverwendbare Muster zum Erstellen von Cloud-nativen Anwendungen

Lehrbuch der Softwaretechnik: Entwurf, Implementierung, Installation und Betrieb

M-WISE

Grundlagen, Architektur und Implementierung in C/C++, Java und C#

Entwurfsprinzipien und Konstruktionskonzepte der Softwaretechnik

Websites mit Struts 1.2 & 1.3 und Ajax effizient entwickeln

Im Mittelpunkt dieses Buches steht der Entwurf von Softwarearchitekturen. Ausgehend von globalen Architekturmustern werden zunächst Einzelaspekte mit Ihren Alternativen behandelt. Damit immer der Bezug zur Realität vorhanden ist, wird eine durchgängige Fallstudie in verschiedenen Varianten zunächst für Einzelaspekte entworfen und implementiert. Neben der Java EE-Plattform wird auch die .NET-Plattform behandelt. Zusätzlich werden die Besonderheiten bei softwareintensiven Systemen dargestellt.

Inhaltsangabe:Einleitung: Diese Diplomarbeit entstand im Zusammenhang mit meiner Tätigkeit als freier Programmierer für ein reales Kundenprojekt. Mein Ein-Personen-Betrieb ist auf die Erstellung von Flash-Anwendungen spezialisiert, wornt eine grundlegende Architekturentscheidung bereits vom Kunden noch vor der eigentlichen Arbeit getroffen wurde. Unabhängig von dieser Entscheidung, die ja immer noch revidiert werden könnte, da es zu Flash ja auch Alternativen gibt (Director, Java-Applets usw.), die der Kunde vielleicht nicht kennt, stellen sich jedoch die allgemeinen Probleme der Softwareentwicklung, die gelöst werden müssen. Problemstellung: Die Arbeit zeigt am Beispiel einer Schach-Community, wie große Softwareprojekte geplant und umgesetzt werden können. Dabei werden die verschiedenen Aspekte quater Softwarearchitektur beleuchtet und am praktischen Beispiel von Kunden, bzw. umgesetzt. Der Zusammenhang zwischen Softwarearchitektur und Vorgehensmodell wird entsprechend dargestellt. Das Projekt verwendet hauptsächlich objektorientierte Techniken und Methoden. Verwendete Technologien sind Flash, der Flash Media Server, Java, Servlets, Tomcat und eine PostgreSQL-Datenbank. Die gewählte Modellierungssprache ist UML 2.0. Eine guten Überblick, auch als Einführung in das Thema, bieten die Folien und der Text für das Kolloquium, die als Flash-Dateien verfügbar sind. Inhaltsverzeichnis:Inhaltsverzeichnis: VorwortII Probleme der SoftwareentwicklungII KomplexitätIII WarenzeichenIII VerzeichnisIV InhaltIV AbbildungenVII TabellenVIII CodebeispieleIX AbkürzungenX 1.Einführung1
1.1.Architekturbegriff1
1.2.Problemstellung1
1.5.Anforderungsbasierte Architektur2
1.4Der Softwarearchitekt2
1.5.Leitfadenz3
2.Grundlagen und erste Ansätze4
2.1 Software als System4
2.2Die Phasenmodelle4
2.2.1Wasserfallmodell4
2.2.2Struktmodell5
2.2.4Diamant-Modell5
2.2.3Projekt-Modell6
2.2.3Architekturstrukturen6
2.3.4Client-Server-Modell6
2.3.2n-Tier-Architektur7
2.3.3Rich-Client / Thin-Client-Architektur10
2.4Entwurfsmuster12
2.4.1Adapter-Muster13
2.4.2Observer-Muster14
2.4.3Strategy-Muster15
2.4.4Composite-Muster17
2.4.5MVC-Paradigma18
2.5.Persistenz21
2.5.1Strukturbruch21
2.5.2Persistenzunabhängigkeit21
2.5.3Persistenzschiicht22
2.5.4Modelltransformation22

2.5.5.Persistenzframework22
6.Flash Media I [...]

Konzepte des agilen Management-Framework, das weitestgehend standardisiert sein, damit die Entwickler sich leicht über Architekturen austauschen können. Für den objektorientierten Entwurf haben sich zahlreiche wertvolle Architektur- und Entwurfsmuster herausgebildet. Diese Muster basieren auf objektorientierten Prinzipien wie dem Prinzip der Dependency Inversion. Daher werden zuerst die wichtigsten objektorientierten Prinzipien erklärt. Anschließend wird gezeigt, wie diese objektorientierten Prinzipien in den verschiedenen Architektur- und Entwurfsmustern umgesetzt werden. Alle vorgestellten Muster werden durch lauffähige Beispiele in Java illustriert.

UML 2 erfolgreich einsetzen

Objektorientierte Software-Entwicklung mit der Unified Modeling Language

Struts

Modellierung wissensintensiver Prozesse im Software Engineering

Eine Einführung für Informatiker und Ingenieure: Systeme, Erfahrungen, Methoden, Tools

35 neue Wege zur Verbesserung Ihrer Programme und Entwürfe

ein Pattern-System