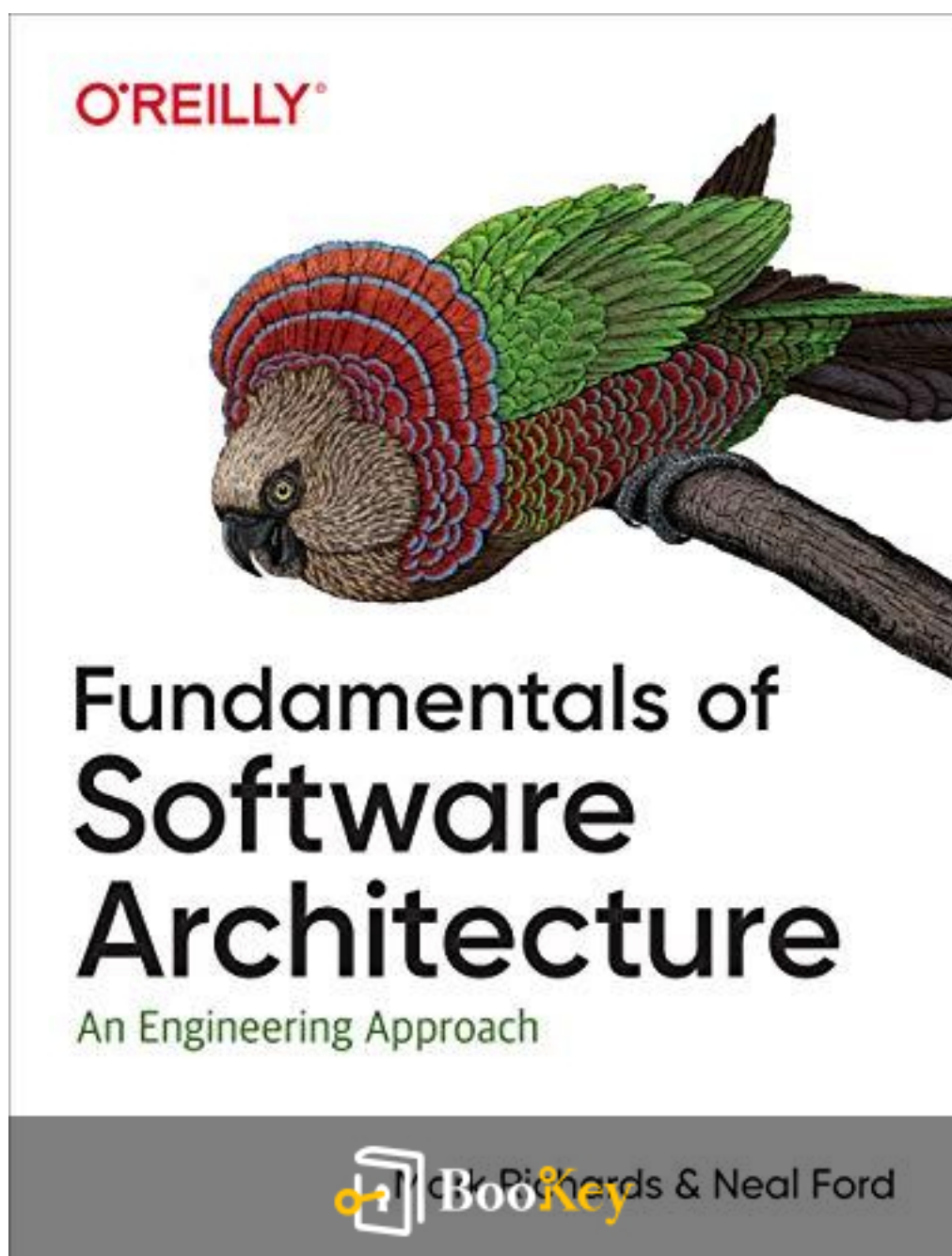


Fundamentos Da Arquitetura De Software PDF (Cópia limitada)

Mark Richards



Teste gratuito com Bookey



Digitalize para baixar

Fundamentos Da Arquitetura De Software Resumo

Princípios de Design para Sistemas Robustos e Escaláveis.

Escrito por Books1

Teste gratuito com Bookey



Digitalize para baixar

Sobre o livro

Bem-vindo ao fascinante mundo de "Fundamentos da Arquitetura de Software", um guia transformador escrito pelo renomado especialista em arquitetura, Mark Richards. Este livro oferece um conjunto de ferramentas essenciais para navegar pelo labirinto das arquiteturas de software, com uma combinação perspicaz de fundamentos teóricos e práticas pragmáticas. Com a expertise de Richards, você descobrirá as sutilezas dos padrões arquiteturais que moldam as soluções de software modernas e mergulhará em estudos de caso do mundo real, cuidadosamente selecionados, que vão além dos limites do aprendizado tradicional. Seja você um arquiteto experiente buscando aprimorar sua habilidade ou um desenvolvedor ansioso para explorar a dimensão arquitetônica do design de software, este livro ilumina o caminho para criar sistemas escaláveis, eficientes e resilientes com clareza e precisão. Prepare-se para embarcar em uma jornada que não apenas aprimora sua competência técnica, mas também inspira uma apreciação mais profunda pela arte e ciência da arquitetura de software.

Teste gratuito com Bookey



Digitalize para baixar

Sobre o autor

Mark Richards é uma autoridade reconhecida mundialmente em arquitetura de software, celebrado por sua abordagem pragmática e esclarecedora na intermediação entre sistemas técnicos complexos e soluções de design intuitivas. Com mais de três décadas de ampla experiência na indústria, Mark ocupou diversas funções de liderança — desde desenvolvedor de software até arquiteto-chefe — em setores variados, incluindo finanças, saúde e governo. É renomado por sua capacidade de descomplicar conceitos arquiteturais intrincados, tornando-se um palestrante e educador procurado em conferências internacionais. Autor publicado, Mark possui um repertório impressionante de obras que exploram as nuances da engenharia de software, incluindo seu trabalho fundamental, "Fundamentos da Arquitetura de Software", co-autorado com Neal Ford. Seu profundo compromisso com o avanço dos princípios fundamentais da arquitetura é sublinhado por sua dedicação em ajudar profissionais a criar sistemas escaláveis e manuteníveis, capazes de se adaptar às demandas em constante evolução do cenário tecnológico.

Teste gratuito com Bookey



Digitalize para baixar



Experimente o aplicativo Bookey para ler mais de 1000 resumos dos melhores livros do mundo

Desbloqueie **1000+** títulos, **80+** tópicos

Novos títulos adicionados toda semana

Product & Brand

 Liderança & Colaboração

 Gerenciamento de Tempo

 Relacionamento & Comunicação

 Estratégia de Negócios

 Criatividade

 Memórias

 Conheça a Si Mesmo

 Psicologia

Empreendedorismo

 História Mundial

 Comunicação entre Pais e Filhos

 Autocuidado

 Mente

Visões dos melhores livros do mundo

amento
pos

Os 7 Hábitos das
Pessoas Altamente
Eficazes



Mini Hábitos



Hábitos Atômicos



O Clube das 5
da Manhã



Como Fazer Amigos
e Influenciar
Pessoas



Com
Não



Teste gratuito com Bookey



Lista de Conteúdo do Resumo

Claro! Aqui está a tradução para o português do "Capítulo 1":

****Capítulo 1****

Se precisar de mais conteúdo para traduzir ou de ajuda com outras partes, é só avisar!: 3. Fundamentos da Arquitetura de Software

Capítulo 2: 7.2. Pensamento Arquitetônico

Capítulo 3: 8. 3. Modularidade

Capítulo 4: Sure! The translation of "10. 5. Identifying Architectural Characteristics" into Portuguese would be:

"10. 5. Identificação das Características Arquitetônicas"

Sure! The translation for "Chapter 5" in Portuguese is "Capítulo 5." If you need any further assistance with translations or anything else, feel free to ask!: 11.6. Medindo e Governando as Características da Arquitetura

Capítulo 6: Certainly! The translation of "12. 7. Scope of Architecture Characteristics" into natural, commonly used Portuguese for readers who enjoy literature would be:

****12.7. Escopo das Características da Arquitetura****

Teste gratuito com Bookey



Digitalize para baixar

Capítulo 7: Sure! The English term "15. 9. Foundations" can be translated into Portuguese as:

"15. 9. Fundamentos"

If you need a more contextual translation or further details, please provide more context about the content surrounding this term!

Capítulo 8: 16. 10. Estilo de Arquitetura em Camadas

Claro! A tradução de "Chapter 9" para o português é "Capítulo 9". Se precisar de mais ajuda com a tradução de textos ou frases, estou à disposição!: 19. 13. Estilo de Arquitetura Baseado em Serviços

Teste gratuito com Bookey



Digitalize para baixar

**Claro! Aqui está a tradução para o português do
"Capítulo 1":**

****Capítulo 1****

**Se precisar de mais conteúdo para traduzir ou de ajuda
com outras partes, é só avisar! Resumo: 3. Fundamentos
da Arquitetura de Software**

Resumo de "Fundamentos da Arquitetura de Software" de Mark
Richards e Neal Ford

"Fundamentos da Arquitetura de Software" é um guia abrangente que
explora os elementos e princípios cruciais para o design e construção de
sistemas de software robustos. Escrito por Mark Richards e Neal Ford,
arquitetos de software experientes, o livro busca fechar a lacuna entre
conceitos teóricos e aplicações práticas no campo da arquitetura de software.

Introdução à Arquitetura de Software

O livro começa com uma introdução ao papel fundamental do arquiteto de
software, destacando a necessidade de equilibrar expertise técnica com
habilidades de liderança. A arquitetura de software envolve entender
requisitos não funcionais, como desempenho, escalabilidade e

Teste gratuito com Bookey



Digitalize para baixar

manutenibilidade, e traduzir esses elementos em designs de sistemas concretos.

Conceitos e Princípios Chave

Richards e Ford mergulham em princípios fundamentais que orientam os arquitetos na tomada de decisões arquitetônicas. Esses princípios incluem a compreensão de estilos arquitetônicos, como arquiteturas monolíticas, microserviços e orientadas a eventos, cada uma com suas vantagens e desafios. Os autores enfatizam a importância de avaliar os trade-offs, uma vez que cada escolha arquitetônica impacta diferentes aspectos do sistema.

Características Arquitetônicas

Os autores categorizam as qualidades que um sistema bem arquitetado deve apresentar, como escalabilidade, que garante que o sistema possa crescer com o aumento das demandas; e flexibilidade, que permite modificações e evolução ao longo do tempo. Eles também ressaltam a importância de compreender requisitos específicos do domínio para criar arquiteturas que atendam precisamente às necessidades únicas de negócios.

Técnicas e Ferramentas

Richards e Ford oferecem insights sobre várias ferramentas e técnicas que

Teste gratuito com Bookey



Digitalize para baixar

auxiliam os arquitetos no design e na avaliação de arquiteturas de software. Eles discutem a relevância da modelagem e documentação como meios de comunicar ideias arquitetônicas. Além disso, os autores introduzem ferramentas de análise crítica, como avaliação de riscos e análise de custo-benefício, para garantir que as decisões estejam alinhadas com os objetivos de negócios.

O Papel do Arquiteto de Software

O livro explora ainda o papel em evolução do arquiteto de software, ressaltando a necessidade de os arquitetos adotarem uma abordagem voltada para o futuro. Eles devem se manter atualizados com os avanços tecnológicos e as tendências emergentes, assegurando que suas soluções permaneçam relevantes e compatíveis com o futuro. Os autores enfatizam a posição do arquiteto como um líder na definição da direção técnica, ao mesmo tempo em que facilita a comunicação entre as partes interessadas.

Aplicações Práticas

Nos capítulos finais, os autores apresentam cenários do mundo real e estudos de caso que ilustram desafios arquitetônicos e soluções implementadas em projetos da indústria. Isso evidencia a aplicação dos princípios discutidos em ambientes práticos e mostra como os arquitetos podem resolver problemas complexos de maneira criativa.



Em resumo, "Fundamentos da Arquitetura de Software" serve como um guia essencial tanto para arquitetos iniciantes quanto para os mais experientes. Ele proporciona uma base sólida em conceitos arquitetônicos e equipa os leitores com as habilidades necessárias para projetar sistemas de software eficientes, escaláveis e adaptáveis.

Teste gratuito com Bookey



Digitalize para baixar

Capítulo 2 Resumo: 7.2. Pensamento Arquitetônico

Capítulo 2 do livro aprofunda o conceito de "Pensamento Arquitetônico", que envolve ver projetos e problemas a partir de uma perspectiva arquitetônica, distinta da visão de um desenvolvedor. Esse conceito é comparado à forma como um meteorologista percebe nuvens de maneira diferente de um artista. O capítulo transmite que o pensamento arquitetônico vai além de simplesmente refletir sobre arquitetura; ele incorpora uma mentalidade e uma abordagem específicas.

O capítulo destaca quatro componentes chave para pensar como um arquiteto. Em primeiro lugar, é necessário distinguir arquitetura de design. O capítulo busca esclarecer a linha muitas vezes nebulosa entre os dois, explicando os papéis e responsabilidades únicos dos arquitetos em comparação aos desenvolvedores. Um arquiteto foca nos componentes estruturais e sistêmicos mais amplos de um projeto, enquanto um designer pode se concentrar nos detalhes mais finos e na implementação.

Em segundo lugar, o capítulo enfatiza a importância de possuir um amplo conhecimento técnico aliado a uma profundidade técnica adequada. Essa dupla expertise permite que os arquitetos identifiquem soluções e oportunidades únicas que podem escapar a outros. Ela incentiva a colaboração com equipes de desenvolvimento, promovendo um ambiente onde a arquitetura se torna um esforço coletivo.



Em terceiro lugar, um arquiteto deve analisar e conciliar eficazmente as trocas entre diferentes soluções e tecnologias. Isso significa entender os pontos fortes e limitações de várias abordagens, tomando decisões informadas que equilibrem esses aspectos em alinhamento com os objetivos do projeto.

Por fim, o capítulo discute a importância de compreender os fatores de negócios e seu impacto nas decisões arquitetônicas. Os arquitetos precisam traduzir os requisitos de negócios em preocupações arquitetônicas, garantindo que suas decisões apoiem os objetivos empresariais mais amplos.

Ao explorar esses quatro aspectos, o capítulo visa oferecer insights sobre a adoção de uma mentalidade arquitetônica, ajudando indivíduos a ver projetos sob a ótica de um arquiteto e cultivando uma abordagem abrangente para desenvolver soluções arquitetônicas robustas.



Pensamento Crítico

Ponto Chave: Reconhecendo e conciliando as compensações nas soluções

Interpretação Crítica: Na sua vida cotidiana, abraçar o conceito de analisar e reconciliar as compensações pode transformar drasticamente a forma como você enfrenta desafios e toma decisões. Imagine cada escolha que você faz como uma balança, onde cada lado possui diferentes benefícios e desvantagens. Ao adotar uma mentalidade que avalia essas compensações, você pode tomar decisões mais informadas e equilibradas. Essa consciência permite que você aprecie a complexidade de cada cenário, entenda seus variados impactos e, em última análise, escolha caminhos que estejam alinhados com suas metas e valores pessoais. Assim como um pensador arquitetônico pesando os prós e contras de soluções técnicas, você pode aplicar essa habilidade para gerir seus recursos, relacionamentos e tempo de forma mais eficaz, levando a uma vida mais harmoniosa e gratificante.

Teste gratuito com Bookey



Digitalize para baixar

Capítulo 3 Resumo: 8. 3. Modularidade

Capítulo 3: Modularidade

Este capítulo aborda o conceito fundamental de modularidade na arquitetura de software, um princípio amplamente discutido, mas muitas vezes mal compreendido ou definido de forma inconsistente. O termo "modularidade" refere-se à prática de organizar o código em unidades distintas e gerenciáveis, chamadas módulos. Esses módulos agrupam funcionalidades relacionadas, promovendo a reutilização e a manutenibilidade nos sistemas de software. Apesar de sua prevalência, a definição de modularidade varia bastante entre fontes e plataformas, gerando confusão.

A citação de Glenford J. Myers enfatiza o foco de longa data nos benefícios da modularidade nas discussões sobre arquitetura de software, mas aponta a falta de orientações sobre como alcançá-la de forma eficaz. Este capítulo busca esclarecer isso ao oferecer uma definição consistente e explorar as aplicações práticas da modularidade no design arquitetônico.

Para os arquitetos, entender as sutilezas da modularidade dentro de sua plataforma de desenvolvimento escolhida é crucial. Ela fundamenta várias ferramentas de análise arquitetônica, como métricas, que medem as propriedades do sistema; funções de adequação, que avaliam a conformidade

Teste gratuito com Bookey



Digitalize para baixar

com os resultados desejados; e visualizações, que fornecem insights sobre a estrutura do sistema. A modularidade serve como um princípio organizador que, quando negligenciado, pode resultar em um sistema caótico, propenso a dificuldades de manutenção e escalabilidade.

Usando uma analogia da física, o capítulo sugere que os sistemas de software são semelhantes a sistemas complexos na natureza, tendendo inerentemente à entropia ou desordem. Nessa analogia, a modularidade fornece uma estrutura para combater essa desordem, organizando os sistemas de maneira gerida e coesa.

No geral, o capítulo destaca a importância de um conceito de modularidade bem definido e aplicado como um pilar da arquitetura de software eficaz, servindo tanto como um princípio orientador quanto como um meio para facilitar o uso de várias ferramentas analíticas. A discussão a seguir ressalta como os arquitetos podem incorporar a modularidade em seus designs para criar sistemas de software robustos, adaptáveis e eficientes.



Capítulo 4: Sure! The translation of "10. 5. Identifying Architectural Characteristics" into Portuguese would be:

"10. 5. Identificação das Características Arquitetônicas"

****Capítulo 5: Extrair Características Arquitetônicas a partir de Preocupações do Domínio****

O capítulo 5 mergulha no processo fundamental de identificação das características arquitetônicas, que são essenciais para o desenvolvimento de uma arquitetura de software robusta ou para a avaliação de uma existente. Conhecidas coloquialmente como “-idades” (como escalabilidade, segurança e desempenho), essas características são cruciais para garantir que a arquitetura esteja alinhada com os objetivos de negócio e os requisitos técnicos.

Para identificar com precisão essas características arquitetônicas, um arquiteto precisa entender bem o problema do domínio. Isso envolve uma colaboração estreita com as partes interessadas para identificar o que é crucial do ponto de vista delas. A contribuição das partes interessadas é inestimável, pois reflete as necessidades práticas e os desafios específicos do domínio.

Os arquitetos podem descobrir características arquitetônicas por meio de

Teste gratuito com Bookey



Digitalize para baixar

várias abordagens. O capítulo destaca três métodos principais: interpretar preocupações do domínio, atender aos requisitos e aproveitar o conhecimento implícito do domínio. Embora as características implícitas tenham sido abordadas em uma seção anterior, este capítulo foca nos dois primeiros métodos.

O processo começa com a tradução das preocupações do domínio em características arquitetônicas. Por exemplo, um arquiteto pode questionar se a prioridade do sistema deve ser a escalabilidade ou se atributos como tolerância a falhas, segurança ou desempenho são mais críticos. Às vezes, todas essas características podem ser essenciais para a arquitetura. Portanto, é crucial entender os objetivos primários e o contexto específico do domínio para traduzir preocupações em “-idades” relevantes. Essas preocupações “traduzidas” então formam a base para a tomada de decisões arquitetônicas informadas.

Um conselho oferecido é simplificar a lista de características arquitetônicas. Arquitetos, juntamente com as partes interessadas do domínio, devem se esforçar para manter uma lista concisa de características essenciais, para evitar uma complexidade esmagadora. Complicar demais a arquitetura ao tentar abordar muitas características ao mesmo tempo é um problema reconhecido, que pode diluir o foco e dificultar a tomada de decisões arquitetônicas eficazes.



Em resumo, o Capítulo 5 enfatiza a importância de se envolver profundamente com as preocupações do domínio e com as contribuições das partes interessadas para destilar um conjunto focado e essencial de características arquitetônicas. Essas características influenciam diretamente as decisões arquitetônicas que ajudam a atender aos requisitos do sistema de forma eficaz e servem como um caminho estruturado para uma implementação bem-sucedida do sistema.

Instale o app Bookey para desbloquear o texto completo e o áudio

Teste gratuito com Bookey





Por que o Bookey é um aplicativo indispensável para amantes de livros

- **Conteúdo de 30min**
Quanto mais profunda e clara for a interpretação que fornecemos, melhor será sua compreensão de cada título.
- **Clipes de Ideias de 3min**
Impulsione seu progresso.
- **Questionário**
Verifique se você dominou o que acabou de aprender.
- **E mais**
Várias fontes, Caminhos em andamento, Coleções...

Teste gratuito com Bookey



Sure! The translation for "Chapter 5" in Portuguese is "Capítulo 5." If you need any further assistance with translations or anything else, feel free to ask! Resumo: 11.6. Medindo e Governando as Características da Arquitetura

****Capítulo 6: Medindo e Governando Características da Arquitetura****

No âmbito dos projetos de software, os arquitetos enfrentam uma vasta gama de características arquitetônicas que abrangem várias dimensões. Entre estas estão aspectos operacionais como desempenho, elasticidade e escalabilidade, bem como preocupações estruturais, como modularidade e capacidade de implantação. Este capítulo se aprofunda na definição clara dessas características e no estabelecimento de mecanismos de governança para gerenciá-las de forma eficaz.

****Medindo Características da Arquitetura****

Um desafio fundamental reside na definição inconsistente das características da arquitetura dentro das organizações. Vários problemas são comuns:

1. ****Não são Física****: Ao contrário da precisão das ciências físicas, as características arquitetônicas muitas vezes carregam significados ambíguos.

Teste gratuito com Bookey



Digitalize para baixar

Por exemplo, conceitos como agilidade ou capacidade de implantação são amplamente utilizados, mas interpretados de maneiras diferentes entre equipes e projetos. Essa ambiguidade surge de perspectivas distintas da indústria, impulsionadas por contextos diferentes ou desalinhamentos acidentais.

2. ****Definições Extremamente Variadas****: Mesmo dentro de uma única organização, departamentos diferentes podem ter entendimentos conflitantes sobre características essenciais como desempenho. Sem uma definição unificada, o diálogo construtivo entre desenvolvedores, arquitetos e equipes de operações se torna desafiador.

3. ****Demasiado Composto****: Muitas características arquitetônicas de alto nível são, na verdade, composições de atributos em menor escala. Por exemplo, a agilidade pode ser decomposta em elementos como modularidade, capacidade de implantação e testabilidade. Reconhecer essa natureza decomponível ajuda a melhorar a medição e a governança.

Para enfrentar esses desafios, o capítulo sugere o desenvolvimento de definições objetivas para as características da arquitetura. Essas definições facilitam uma compreensão comum, permitindo que as organizações se envolvam em conversas significativas e elaborem estratégias de forma eficaz. Ao alinhar perspectivas diferentes, as organizações podem garantir que as decisões de arquitetura sejam coerentes e alinhadas com seus



objetivos gerais. Esses esforços, em última análise, aprimoram o ciclo de vida de desenvolvimento de software ao criar estruturas robustas para medir e governar características arquitetônicas-chave.

Seção	Resumo
Visão Geral	Os arquitetos lidam com várias características da arquitetura, que são cruciais para projetos de software. Essas características abrangem elementos operacionais como desempenho e escalabilidade, além de elementos estruturais como modularidade. Uma gestão eficaz é necessária por meio de definições claras e mecanismos de governança.
Desafios na Medição	Não são Física: As características da arquitetura, ao contrário das leis físicas, são frequentemente definidas de maneira ambígua, levando a interpretações variadas entre equipes e projetos. Definições Extremamente Variadas: Existem definições conflitantes dentro das organizações, complicando a comunicação e a colaboração entre desenvolvedores, arquitetos e equipes de operações. Demasiado Compostas: Características de alto nível são compostas por atributos menores. Decompor essas características ajuda a melhorar a medição e a governança.
Solução	Devem ser desenvolvidas definições objetivas para as características da arquitetura. Esse entendimento comum possibilita discussões significativas, alinhamento estratégico e decisões arquitetônicas coerentes com os objetivos organizacionais. Tal alinhamento melhora a governança e a medição das características arquitetônicas ao longo do ciclo de vida do software.



Capítulo 6 Resumo: Certainly! The translation of "12. 7. Scope of Architecture Characteristics" into natural, commonly used Portuguese for readers who enjoy literature would be:

****12.7. Escopo das Características da Arquitetura****

Capítulo 7 do livro intitulado "Características da Arquitetura e seu Escopo" explora a natureza em evolução das características da arquitetura de software, com um foco especial em como o escopo se deslocou de uma perspectiva de sistema inteiro para níveis mais granulares. Tradicionalmente, as discussões sobre características de arquitetura, como escalabilidade, ocorriam em nível de sistema. Essa perspectiva ampla era adequada quando a maioria dos sistemas era monolítica. No entanto, o surgimento das práticas de engenharia modernas e dos estilos arquitetônicos, notavelmente os microserviços, exigiu uma reavaliação dessa visão.

Os microserviços, uma abordagem para desenvolver sistemas de software em pequenos módulos independentes, ressaltam como as características da arquitetura agora precisam ser consideradas em níveis mais localizados. Cada microserviço pode ter suas próprias preocupações em relação à escalabilidade, desempenho e segurança, separadas do sistema mais amplo. Essa mudança personifica como os axiomas no desenvolvimento de software podem se tornar obsoletos à medida que o campo evolui.



Ao escreverem seu livro "Construindo Arquiteturas Evolutivas", os autores identificaram uma lacuna na medição da evolutividade estrutural de diferentes estilos de arquitetura. As métricas existentes forneciam insights sobre detalhes em nível de código, mas não consideravam dependências estruturais mais amplas, como bancos de dados ou serviços externos, que impactam significativamente as características da arquitetura. Esses elementos, externos à base de código, continuam a desempenhar um papel crucial nos aspectos operacionais da arquitetura de um sistema.

Por exemplo, o desempenho e a elasticidade de um sistema não são ditados apenas por um código bem projetado, mas também são fortemente influenciados por componentes externos, como bancos de dados. Essa percepção destaca a importância de se considerar dependências além da base de código imediata ao avaliar as características da arquitetura. Portanto, o capítulo defende uma compreensão e abordagem mais nuançadas para avaliar e monitorar as características da arquitetura no contexto de sistemas modernos e distribuídos.



Pensamento Crítico

Ponto Chave: Avaliação granular das características da arquitetura

Interpretação Crítica: Imagine as mudanças na sua vida vistas como componentes arquitetônicos dentro de um sistema complexo mais amplo. Assim como a arquitetura de software evoluiu de estruturas monolíticas para microserviços, seu crescimento pessoal e adaptações exigem uma avaliação de cada elemento da sua vida em um nível micro. Em vez de perceber seu bem-estar, produtividade ou relacionamentos como entidades singulares e abrangentes, considere-os módulos distintos - cada um com desafios e requisitos únicos. Ao focar nas relações e dependências entre essas áreas da vida, você pode projetar uma vida mais resiliente e adaptável. Reconheça como as influências externas — semelhantes a bancos de dados em software — afetam a arquitetura da sua vida. Talvez sejam relacionamentos ou obrigações externas que impactam seu desempenho ou elasticidade emocional. Use este ponto-chave para inspirar uma mudança de perspectiva, priorizando tanto a interconexão quanto a independência dos vários projetos da vida, aprimorando tanto a resiliência quanto a flexibilidade.

Teste gratuito com Bookey



Digitalize para baixar

Capítulo 7 Resumo: Sure! The English term "15. 9. Foundations" can be translated into Portuguese as:

"15. 9. Fundamentos"

If you need a more contextual translation or further details, please provide more context about the content surrounding this term!

Capítulo 9: Fundamentos

No Capítulo 9, intitulado "Fundamentos", o foco está nos estilos arquitetônicos, frequentemente chamados de padrões de arquitetura, que definem uma maneira estruturada de organizar componentes em um sistema de software. Esses estilos servem como uma abreviação para arquitetos experientes, similar aos padrões de design no desenvolvimento de software, ao encapsular relações complexas e características da arquitetura sob um nome facilmente reconhecível. A familiaridade com esses estilos é essencial para os arquitetos, pois permite que se comuniquem de forma eficiente sobre detalhes e conceitos arquétipos intrincados.

Um estilo de arquitetura abrange a topologia e as características padrão de um sistema, delineando tanto as vantagens quanto as desvantagens

Teste gratuito com Bookey



Digitalize para baixar

potenciais. Por exemplo, ao se referir a um "monólito em camadas", os arquitetos reconhecem a organização da estrutura, suas características favoráveis como a manutenibilidade e a facilidade de testes, além de possíveis problemas, como a escalabilidade reduzida. Nomes como esses transmitem um conjunto abrangente de diretrizes implícitas que orientam modelos de implantação, estratégias de gerenciamento de dados e muito mais.

O capítulo sugere que o restante da Parte II irá se aprofundar em padrões arquitetônicos modernos comuns. No entanto, entender certos padrões fundamentais é crítico, pois eles frequentemente formam a espinha dorsal de sistemas mais complexos. Esses padrões fundamentais frequentemente reaparecem ao longo da história da arquitetura de software, proporcionando uma abordagem eficaz para organizar não apenas o código, mas também as implantações e vários aspectos arquitetônicos.

Um exemplo de um padrão fundamental é a arquitetura em camadas, que separa as preocupações com base na funcionalidade em diferentes camadas. Essa separação otimiza a gerenciabilidade e a clareza do código, permitindo que cada camada trate de tarefas distintas. Apesar de ser um conceito fundamental, padrões como esses servem como blocos de construção para estilos arquitetônicos mais complexos e nuançados encontrados no design de software moderno. O capítulo estabelece a base para explorar esses estilos intrincados, fundamentando os leitores nesses princípios arquitetônicos



essenciais.

Teste gratuito com Bookey



Digitalize para baixar

Capítulo 8: 16. 10. Estilo de Arquitetura em Camadas

Capítulo 10 explora o estilo de arquitetura em camadas, também conhecido como arquitetura n-tier, que é um framework comum para estruturar aplicações. Esse estilo arquitetônico é popular devido à sua simplicidade e custo-benefício. Ele se alinha bem à Lei de Conway, que sugere que os designs de sistemas muitas vezes refletem as estruturas de comunicação das organizações que os criam. Na prática, isso significa que os papéis presentes em muitas equipes de desenvolvimento—desenvolvedores de interface do usuário (UI), desenvolvedores de backend, desenvolvedores de regras e administradores de banco de dados—se encaixam perfeitamente na estrutura de uma arquitetura em camadas, tornando-a uma escolha natural para muitas empresas.

A arquitetura em camadas é organizada em camadas horizontais, cada uma com uma responsabilidade distinta, como apresentação ou lógica de negócios. Essa abordagem estruturada não só facilita a clara delimitação de papéis, mas também apoia a manutenção e a escalabilidade. No entanto, essa arquitetura não está isenta de armadilhas. Ela pode levar a anti-padrões arquitetônicos, como “arquitetura por implicação”, onde a arquitetura surge implicitamente em vez de ser definida de forma explícita, e “arquitetura acidental”, onde a arquitetura evolui de maneira desordenada sem um planejamento deliberado.



O capítulo sugere que, se uma equipe de desenvolvimento não tem certeza sobre o estilo de arquitetura que está utilizando—ou se inicia o desenvolvimento sem uma arquitetura definida—é provável que esteja adotando uma arquitetura em camadas por padrão. A familiaridade dessa arquitetura a torna um ponto de partida prático para muitas equipes, especialmente aquelas que utilizam metodologias ágeis, que favorecem a adaptabilidade e o desenvolvimento iterativo. Apesar de seu uso generalizado, compreender as desvantagens potenciais e garantir que a arquitetura esteja alinhada com os objetivos organizacionais é crucial para uma implementação bem-sucedida. O capítulo incentiva uma abordagem deliberada na seleção da arquitetura, considerando tanto as necessidades atuais quanto a escalabilidade futura para evitar armadilhas comuns e apoiar efetivamente o desenvolvimento de aplicações.

Instale o app Bookey para desbloquear o texto completo e o áudio

Teste gratuito com Bookey





App Store
Escolha dos Editores



22k avaliações de 5 estrelas

Feedback Positivo

Afonso Silva

...cada resumo de livro não só
..., mas também tornam o
...divertido e envolvente. O
...tizou a leitura para mim.

Fantástico!



Estou maravilhado com a variedade de livros e idiomas
que o Bookey suporta. Não é apenas um aplicativo, é
um portal para o conhecimento global. Além disso,
ganhar pontos para caridade é um grande bônus!

Brígida Santos

F



O
só
o
O

na Oliveira

...correr as
...ém me dá
...omprar a
...ar!

Adoro!



Usar o Bookey ajudou-me a cultivar um hábito de
leitura sem sobrecarregar minha agenda. O design do
aplicativo e suas funcionalidades são amigáveis,
tornando o crescimento intelectual acessível a todos.

Duarte Costa

Economiza tempo!



O Bookey é o meu apli
crescimento intelectual
perspicazes e lindame
um mundo de conheci

Aplicativo incrível!



Eu amo audiolivros, mas nem sempre tenho tempo para
ouvir o livro inteiro! O Bookey permite-me obter um resumo
dos destaques do livro que me interessa!!! Que ótimo
conceito!!! Altamente recomendado!

Estevão Pereira

Aplicativo lindo



Este aplicativo é um salva-vidas para
de livros com agendas lotadas. Os re
precisos, e os mapas mentais ajudar
o que aprendi. Altamente recomend

Teste gratuito com Bookey



Claro! A tradução de "Chapter 9" para o português é "Capítulo 9". Se precisar de mais ajuda com a tradução de textos ou frases, estou à disposição! Resumo: 19. 13. Estilo de Arquitetura Baseado em Serviços

****Capítulo 13: Estilo de Arquitetura Baseada em Serviços****

A arquitetura baseada em serviços é frequentemente vista como uma abordagem pragmática e flexível para a construção de sistemas de software. Ela funciona como um híbrido do estilo de arquitetura de microserviços, oferecendo muitos de seus benefícios, mas sem o mesmo nível de complexidade e custo associado a microserviços completos ou arquiteturas orientadas a eventos. Isso a torna particularmente atraente para aplicações empresariais que requerem um equilíbrio entre as vantagens de sistemas distribuídos e uma complexidade gerenciável.

****Visão Geral da Topologia****

A arquitetura baseada em serviços utiliza uma topologia distribuída em camadas macro. Essa estrutura é caracterizada por componentes distintos, incluindo uma interface de usuário separadamente implantada, serviços de granularidade mais ampla implantados de forma independente, e um único banco de dados monolítico. Essa abordagem em camadas permite

Teste gratuito com Bookey



Digitalize para baixar

flexibilidade na implantação e escalabilidade dos serviços, enquanto mantém uma integração mais simples com uma fonte de dados unificada.

****Implantação de Serviços e Características****

Nesta arquitetura, os serviços, frequentemente chamados de serviços de domínio, são projetados para encapsular partes específicas de um aplicativo de forma independente. Esses serviços são de granularidade mais ampla, o que significa que abrangem funcionalidades mais amplas e podem operar de forma autônoma. Apesar de serem implantados de forma independente, eles não exigem a containerização que os microserviços requerem; podem ser implantados utilizando métodos tradicionais, como arquivos EAR ou WAR. Isso os torna mais fáceis de gerenciar e reduz a sobrecarga operacional.

Uma arquitetura típica baseada em serviços terá entre quatro e doze serviços dentro de um único contexto de aplicação, com uma configuração média composta por cerca de sete serviços. Esse número reflete um equilíbrio que oferece tanto modularidade quanto simplicidade, permitindo uma manutenção e escalabilidade fáceis.

Em resumo, a arquitetura baseada em serviços oferece um equilíbrio entre a independência modular dos serviços e a gestão simplificada de um sistema monolítico. É uma escolha atraente para empresas que buscam os benefícios de arquiteturas distribuídas sem a complexidade e a demanda de recursos

Teste gratuito com Bookey



Digitalize para baixar

dos microserviços. Através de sua estrutura flexível e requisitos de gestão mais simples, ela apoia a entrega e evolução eficientes de aplicações em diversos cenários empresariais.

Teste gratuito com Bookey



Digitalize para baixar

Pensamento Crítico

Ponto Chave: Equilibrando Complexidade e Simplicidade Através de Arquitetura Baseada em Serviços

Interpretação Crítica: Imagine entrar em uma vida onde os desafios complexos não te sobrecarregam. Você equilibra graciosamente a intrincada com a simplicidade, assim como a arquitetura baseada em serviços equilibra as complexidades dos sistemas distribuídos com a simplicidade dos sistemas monolíticos. Ao compartimentar suas tarefas e responsabilidades em seções autônomas e gerenciáveis, você se permite a flexibilidade para inovar e se adaptar sem ser sobrecarregado. Assim como esse estilo de arquitetura oferece soluções pragmáticas ao integrar flexibilidade com facilidade de gerenciamento, você pode implementar esse equilíbrio em sua vida. Divida seus objetivos em partes focadas, mas amplas, cada uma trabalhando autonomamente em harmonia, ajudando você a crescer e expandir suas aspirações pessoais sem a complexidade desnecessária. Adote uma mentalidade baseada em serviços e misture empreitadas ambiciosas com sistemas estrategicamente posicionados que garantam que você permaneça aterrado e sob controle, pronto para se adaptar a cada novo desafio e oportunidade que surgir em seu caminho.

Teste gratuito com Bookey



Digitalize para baixar