
Introdução à Arquitetura de Software

Prof. Pedro Henrique Dias Valle

Adaptado de:
Profa. Elisa Yumi Nakagawa
Prof. Valdemar Neto

Roteiro

- Introdução
- Definições de Arquitetura de Software
- Conceitos
 - Padrão Arquitetural
 - Visão e Ponto de Vista
 - Decisão Arquitetural
 - Outros

Introdução

- Software é um produto de Engenharia;
- Entretanto, software é um produto abstrato e não tangível;
- Consequentemente, visualizar o "formato" do software é uma atividade complexa;
 - Adicionalmente, mesmo em outras engenharias, não se tem o formato do produto até que ele esteja construído, a não ser pelos modelos (maquete, etc.)

Introdução

- Paralelamente, é difícil construir qualquer coisa cujo formato não seja conhecido;
- Logo, a definição de arquitetura para um produto de engenharia é uma atividade essencial para garantir a qualidade do produto;

Introdução

- Na Arquitetura Civil, arquitetura diz respeito à "forma" das construções:
 - Neoclássica;
 - Barroca;
 - Gótica; etc.

Fonte:
<http://thaa2.files.wordpress.com/2009/07/arquitetura-neoclassica-2.jpg>



Introdução



Fonte:

http://oglobo.globo.com/blogs/arquivos_upload/2009/01/129_302-Francisco,%20Ouro%20Preto.jpg

Fonte:

http://3.bp.blogspot.com/-_hkObXOms_o/T6sBYOwUBDI/AAAAAAAAAEk/d5F3oxSqDhs/s1600/4390-g.jpg



Introdução



Catedral Gótica de Milão.

Fonte:

<https://historiadomundo.uol.com.br/idade-media/inovacao-da-arquitetura-gotica.htm>

Definições

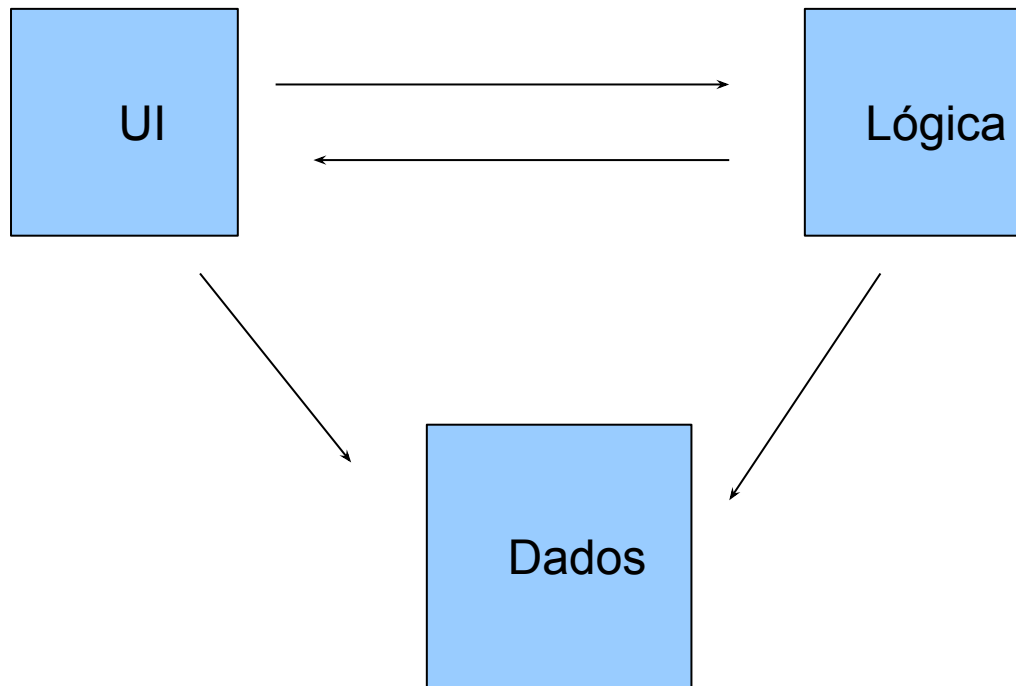
- Afinal, o que é uma Arquitetura de Software?

Definições

- Afinal, o que é uma Arquitetura de Software?
 - Primeira coisa a saber: não há definição consensual;
 - Informal:
 - Diz respeito à forma como porções de código são agrupadas em partes específicas, e como estas partes interagem entre si para culminar no funcionamento do software.

Definições

- Afinal, o que é uma Arquitetura de Software?
 - Informalmente:



Definições

- Martin Fowler:



Editor: Martin Fowler • ThoughtWorks • fowler@acm.org

Who Needs an Architect?

Martin Fowler

Wandering down our corridor a while ago, I saw my colleague Dave Rice in a particularly grumpy mood. My brief question caused a violent statement, "We shouldn't interview anyone who has 'architect' on his resume." At first blush, this was an odd turn of phrase, because we usually introduce Dave as one of our leading architects.



The reason for his title schizophrenia is the fact that, even by our industry's standards, "architect" and "architecture" are terribly overloaded words. For many, the term "software architect" fits perfectly with the smug controlling image at the end of *Matrix Reloaded*. Yet even in firms that have the greatest contempt for that image, there's a vital role for the technical leadership that an architect such as Dave plays.

What is architecture?

When I was fretting over the title for *Patterns of Enterprise Application Architecture* (Addison-Wesley, 2002), everyone who reviewed it agreed that "architecture" belonged in the title. Yet we all felt uncomfortable defining the word. Because it was my book, I felt compelled to take a stab at defining it.

My first move was to avoid fuzziness by just letting my cynicism hang right out. In a sense, I define *architecture* as a word we use when we want to talk about design but want to puff it up to make it sound important. (Yes, you can imagine a similar phenomenon for ar-

chitect.) However, as so often occurs, inside the blighted cynicism is a pinch of truth. Understanding came to me after reading a posting from Ralph Johnson on the Extreme Programming mailing list. It's so good I'll quote it all.

A previous posting said

The RUP, working off the IEEE definition, defines architecture as "the highest level concept of a system in its environment. The architecture of a software system [at a given point in time] is its organization or structure of significant components interacting through interfaces, those components being composed of successively smaller components and interfaces."

Johnson responded:

I was a reviewer on the IEEE standard that used that, and I argued uselessly that this was clearly a completely bogus definition. There is no highest level concept of a system. Customers have a different concept than developers. Customers do not care at all about the structure of significant components. So, perhaps on architecture is the highest level concept that developers have of a system in its environment. Let's forget the developers who just understand their little piece. Architecture is the highest level concept of the expert developers. What makes a component significant? It is significant because the expert developers say so.

So, a better definition would be "In most successful software projects, the expert developers working on that project have a shared understanding of the

Fonte:
<http://martinfowler.com/ieeeSoftware/whoNeedsArchitect.pdf>

Acesso: setembro de 2024

Definições

- Fowler

- "O conceito de mais alto nível que os desenvolvedores têm de um sistema [...] em seu ambiente";
- "Uma consciência compartilhada do projeto (*design*) do sistema [...] e inclui como o sistema é dividido em componentes e como estes componentes interagem através de interfaces."
- "O conjunto de decisões de projeto que devem ser feitas de forma precoce em um projeto".

Definições

- Clements

- "O conjunto de estruturas necessárias para compreender o sistema, e compreende os elementos, relações entre eles, e propriedades de ambos."
- "Arquitetura de Software (AS) é composta dos elementos, conexões, e relações entre eles, e, geralmente, de alguns outros aspectos, tais como configuração, restrições, semântica, detalhes de legislação, análises, e propriedades; ou o *rationale*, requisitos, e necessidades dos stakeholders"

Definições

- Mary Shaw e David Garlan
 - A descrição de elementos a partir dos quais sistemas são construídos, interações entre estes elementos, padrões que guiam sua composição, e restrições sobre estes padrões.
 - Elementos (componentes) neste contexto são clientes e servidores, banco de dados, filtros, camadas, etc.

Definições

- SWEBOK V3
 - O conjunto de estruturas necessário para compreender o sistema, que compreende elementos de software, a relação entre eles, e as propriedades de ambos.

Definições

- Resumindo:
 - Clements et. Al: "“architecture is what you get before you start adding detail to the design.”"
 - Arquitetura de Software diz respeito ao "formato" do software, às partes principais do software, como interação, suas interfaces, e todos os elementos envolvidos no funcionamento e ambientação do software.

Vantagens em Definir uma Arquitetura de Software

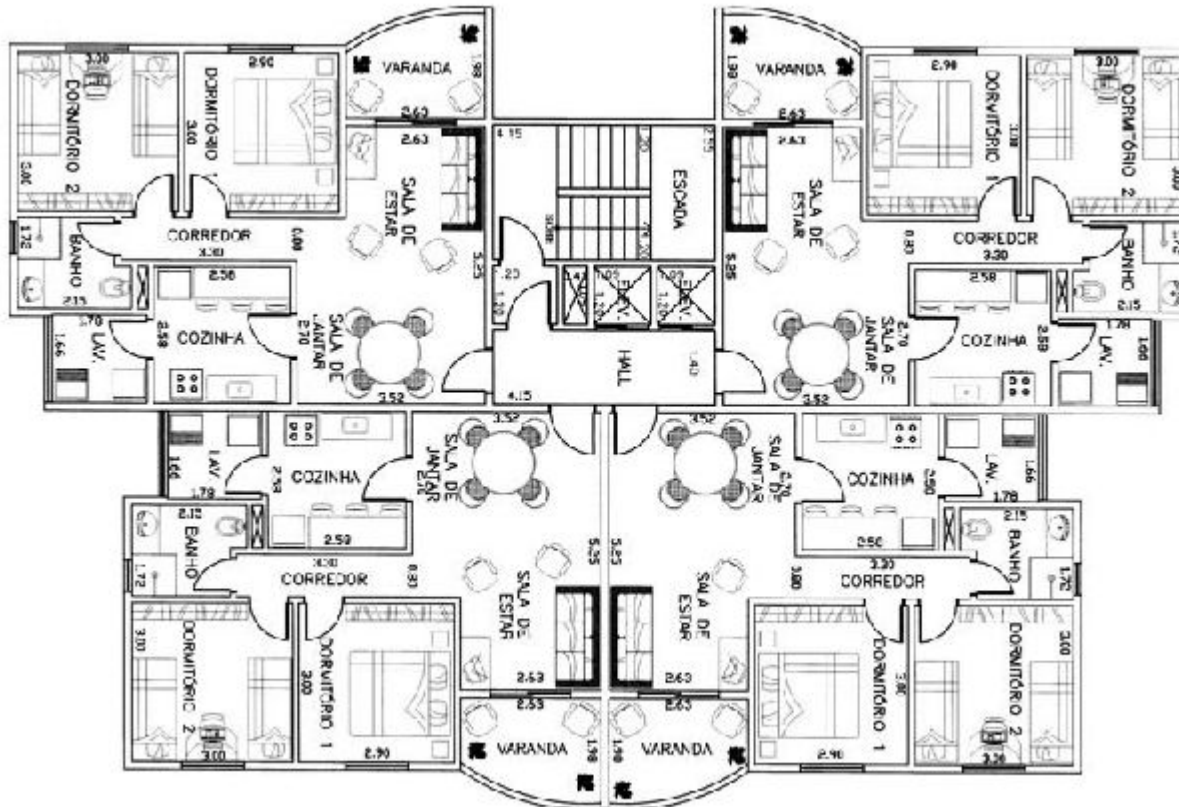
- Manutenibilidade;
- Reúso;
- Encapsulamento;
- Rastreabilidade;
- Qualidade;
- Estabelece vocabulário para diálogo;

Conceitos

- Visão e Pontos de Vista
- Decisão Arquitetural
- Padrão Arquitetural
- Concerns
- Atributos de Qualidade
- Linguagem de Descrição Arquitetural
- Reúso

Visões

- Analogia com Prédio (Visão da Planta)

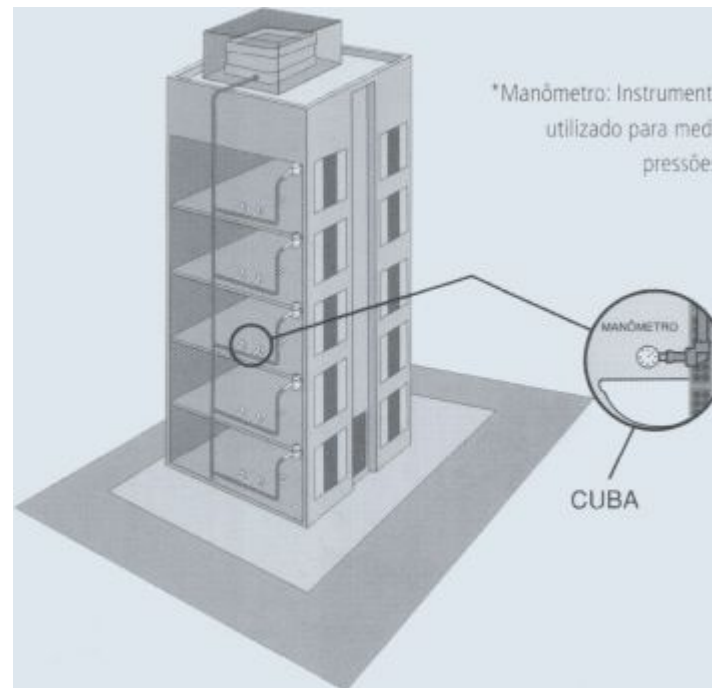


Fonte:

<http://www.conexaolivre.com.br/arquitetura/projetos.grupo2/slides/34-%20Planta%20Edificio%20Sao%20Mauro.jpg>

Visões

- Analogia com Prédio (Visão Hidráulica)



Fonte:

http://www.renatomassano.com.br/dicas/residencial/conceitos_fundamentais/sistema_predial_02.jpg

Visões

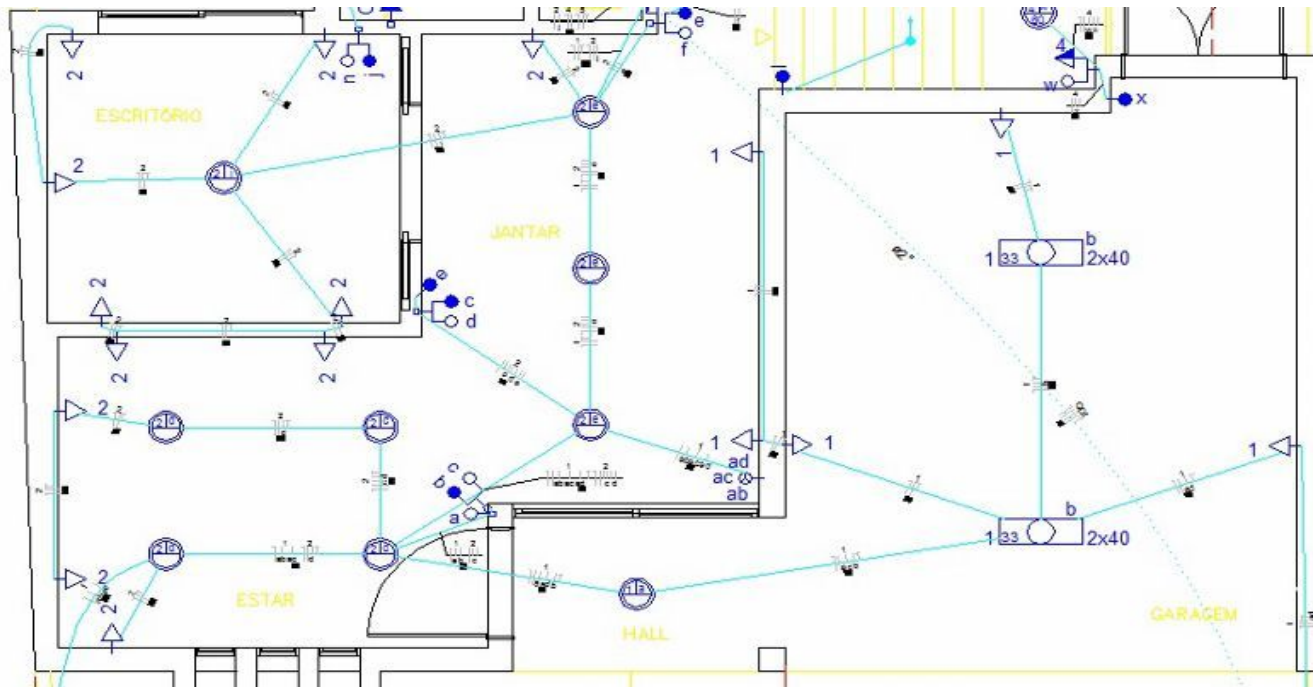
- Analogia com Prédio (Visão Externa)



Fonte: http://1.bp.blogspot.com/-3iQRgm_3-bM/UiSqkA6RPpI/AAAAAAAAACYI/dy7brmlrIGQ/s1600/maquete_predio_acrop_.jpg

Visões

- Analogia com Prédio (Visão do Projeto Elétrico)



Fonte:

<http://www.andreiafigueiredo.com.br/portfolio/Servicos/slides/ELETRICO.jpg>

Visões

- Todas as visões juntas formam a arquitetura do prédio.
- Sozinha, nenhuma delas é capaz de representar o prédio por inteiro.
- Mas juntas, elas formam toda a estrutura do prédio.

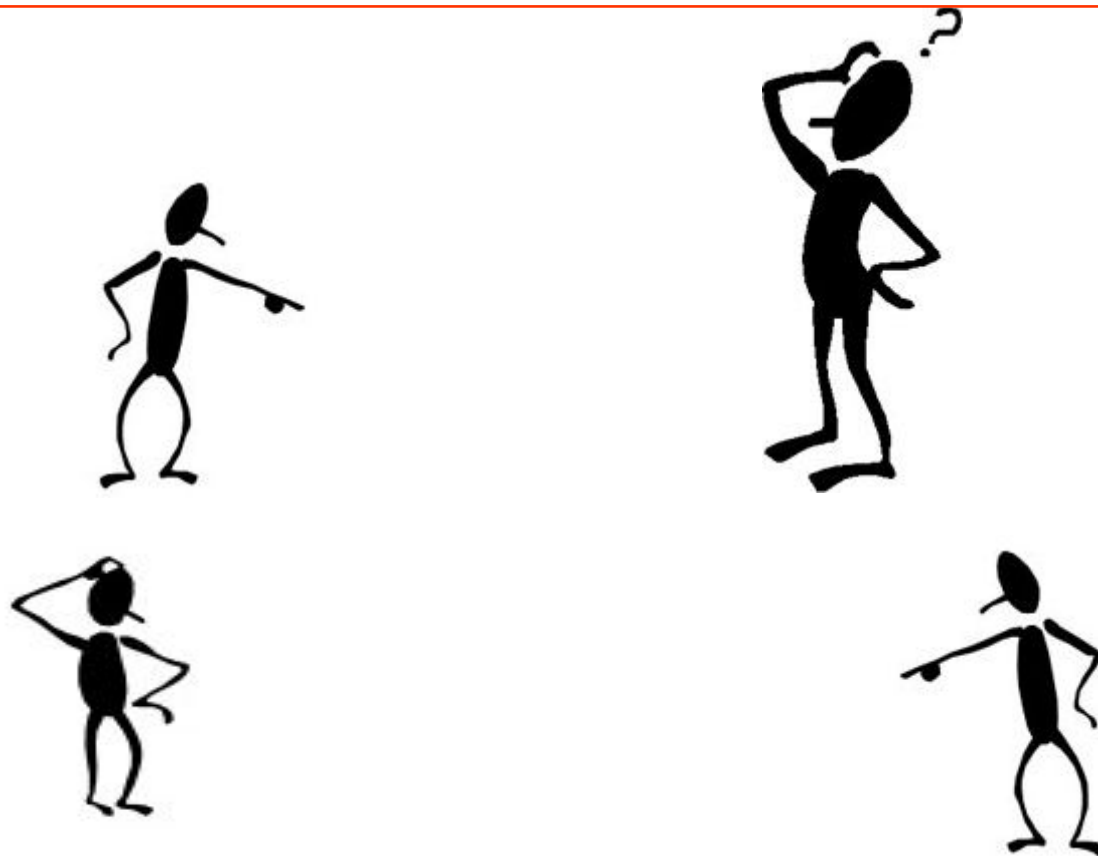
Visões

- Em software, existem também vários modelos que representam várias visões do software como:
 - Visão de Estrutura (principais componentes, classes, etc.);
 - Visão de Comportamento (interação entre elementos, fluxo de informações, etc.);
 - Dentre outros.
- O importante é ser o mais exaustivo possível de modo a representar o software em sua integridade.

Ponto de Vista

- Um ponto de vista origina uma visão.
- O engenheiro elétrico vê o prédio sob a ótica de projeto elétrico; o engenheiro hidráulico sob a ótica de hidráulica, e por aí vai. E cada um gera um modelo diferente.
- Papéis diferentes olhando para um mesmo objeto geram modelos diferentes sobre diferentes aspectos do mesmo objeto observado (ponto de vista gera visão)

Ponto de Vista



Ponto de Vista

Visão 1



Visão 3



Visão 2



Visão 4



Visões

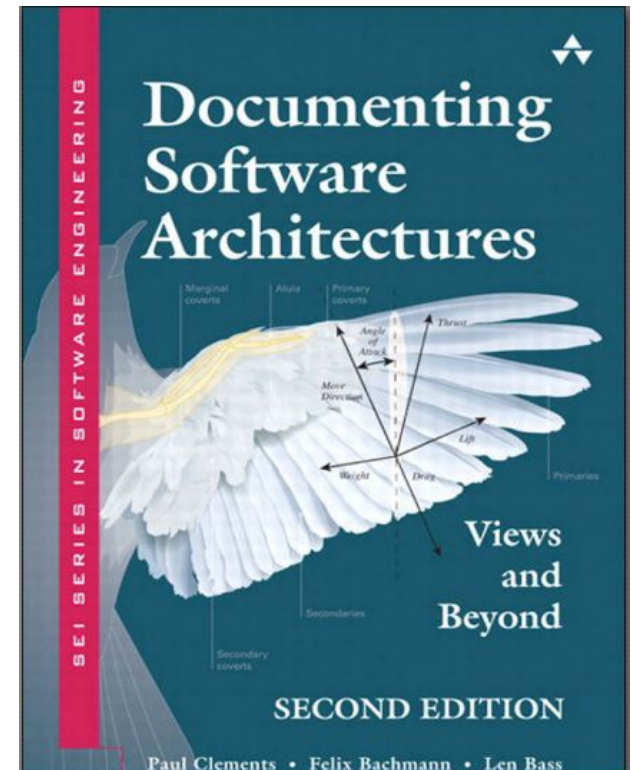
- Mas qual a quantidade ideal?
- E quais tipos de visões usar?
- E quais visões em específico?

Visões

- Sugestões da Literatura:
 - 4+1 Views
 - Views and Beyond (Clements)

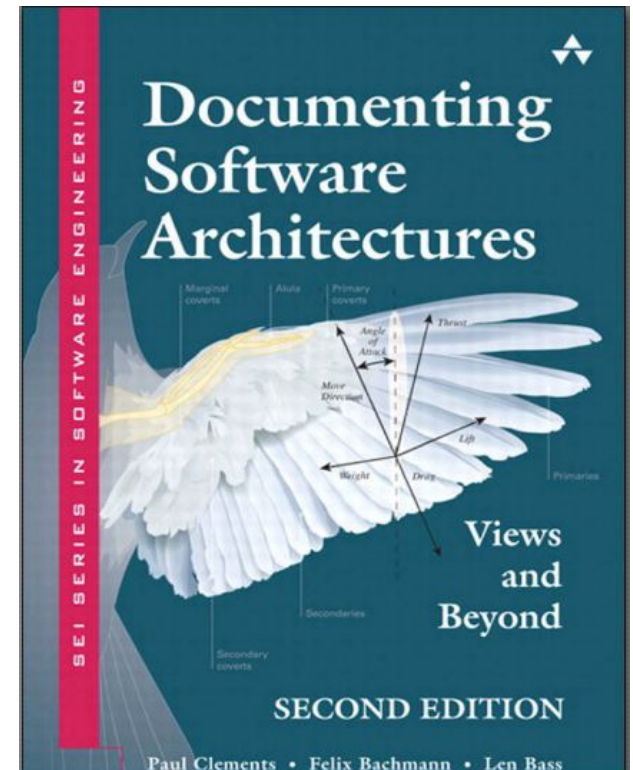
Decisão Arquitetural

- Decisões que permitem ao sistema contemplar seus atributos de qualidade e outros requisitos.
- Ex.:
 - Qual o SGBD para garantir
 - tempo de resposta x ms?
 - Qual organização de componentes para garantir
 - segurança, ou tolerância a falhas (sistemas críticos)?



Decisão Arquitetural

- Como registrar estas decisões arquiteturais?
- Possibilidades:
 - Informal;
 - Formal;
 - Ontologias (Clements et. al)



Padrão Arquitetural

- Padrão (Erich Gamma):
 - Uma solução sistemática, repetível, e adaptável para um problema recorrente.
 - Envolve (uma trinca + o nome):
 - Problema recorrente;
 - Contexto;
 - Solução padronizada.

Padrão Arquitetural

Padrão (Erich Gamma):

- Exemplo:

- Padrão (de Projeto) Singleton:

- Problema: Criar um objeto único dentro de um sistema (exemplo: objeto representando uma conta bancária);
- Contexto: Banco;
- Solução: tornar o método de criação estático e final; construtor private; e invocar criação com construtor de dentro do método.

Padrão Arquitetural

- Problema, Contexto, e Solução para criação de arquiteturas de software;
- Exemplo: Camadas, Componentes, SOA, Cliente-Servidor, etc.
- Também chamados de Estilos Arquiteturais.

Conceitos

- Visão e Pontos de Vista
- Decisão Arquitetural
- Padrão Arquitetural
- Concerns
- Atributos de Qualidade
- Linguagem de Descrição Arquitetural
- Framework Arquitetural
- Reúso

Concerns

- Concerns = Interesses;
- Separação de Interesses é uma grande recomendação de *design* de sistemas;
- Um concern é uma parte do problema que queremos tratar como uma unidade conceitual única na solução do software.
- Definir uma arquitetura de software está inerentemente ligada à tarefa de separação de concerns.
- A Separação de concerns tenta resolver as limitações da cognição humana ao lidar com a complexidade do software.

Concerns

- Os sistemas de software complexos devem ser decompostos em unidades modulares menores e claramente separadas, cada uma lidando com um único concern.
- Exemplos de concern:
 - Persistência;
 - Concorrência;
 - Interação Humano-Computador;
 - Lógica;
 - Negócio;

Concerns

- Orientação a Aspectos é um exemplo de uma técnica utilizado para encapsulamento de interesses (concerns) chamados de transversais;
- Transversais = espalhados no código, repetidos;
- Exemplo:
 - Concorrência;
 - Logging;
 - Tratamento de Exceções;
 - Persistência;

Concerns

- Isolar concerns é um trabalho inerente à definição de Arquitetura de Software.

Conceitos

- Visão e Pontos de Vista
- Decisão Arquitetural
- Padrão Arquitetural
- Concerns
- Atributos de Qualidade
- Linguagem de Descrição Arquitetural
- Framework Arquitetural
- Reúso

Atributos de Qualidade

- Requisitos Não-Funcionais que são utilizados para aferir a qualidade de um sistema;
- Exemplos:
 - Desempenho
 - Disponibilidade
 - Modificabilidade
 - Segurança
 - Testabilidade
 - Usabilidade

Atributos de Qualidade

- Em geral, a definição de uma arquitetura auxilia o engenheiro no cumprimento de atributos de qualidade.
- Exemplo:
 - Tolerância a Falhas pode ser conseguida através de Redundância de unidades de processamento, o que faz parte da definição da arquitetura;
 - Escolha de tecnologias influenciam no desempenho;

Atributos de Qualidade

- Em geral, a definição de uma arquitetura auxilia o engenheiro no cumprimento de atributos de qualidade.
- Exemplo:
 - Disponibilidade está ligada a Redundância também;
 - Estas estratégias são chamadas de **táticas arquiteturais**.

Conceitos

- Visão e Pontos de Vista
- Decisão Arquitetural
- Padrão Arquitetural
- Concerns
- Atributos de Qualidade
- Linguagem de Descrição Arquitetural
- Framework Arquitetural
- Reúso

Linguagem de Descrição Arquitetural

- ADL: Architectural Description Language;
- Uma linguagem específica de domínio para especificar uma arquitetura;
- Em geral, em termos de componentes, conectores, e comportamento;
- Exemplo: Pi-ADL, ACME, etc.

Linguagem de Descrição Arquitetural

- Linguagens de propósito geral também costumam ser utilizadas para especificar arquitetura;
- Pensando em UML, quais diagramas poderiam ser usados para modelar:
 - Estrutura?
 - Comportamento?
 - Implantação?

Linguagem de Descrição Arquitetural

- Linguagens de propósito geral também costumam ser utilizadas para especificar arquitetura;
- Pensando em UML, quais diagramas poderiam ser usados para modelar:
 - Estrutura? Componentes, Pacotes.
 - Comportamento? Sequência, Atividades, Estados.
 - Implantação? Implantação.

Linguagem de Descrição Arquitetural

- Quais outras linguagens de propósito geral, na falta de uma ADL?
 - Comportamento
 - Estrutura

Linguagem de Descrição Arquitetural

- Quais outras linguagens de propósito geral, na falta de uma ADL?
 - Comportamento: BPMN, SPEM, etc.
 - Estrutura: SySML, etc.

Conceitos

- Visão e Pontos de Vista
- Decisão Arquitetural
- Padrão Arquitetural
- Concerns
- Atributos de Qualidade
- Linguagem de Descrição Arquitetural
- **Framework Arquitetural**
- **Reúso**

Framework Arquitetural

- Conjunto de recomendações para criação de uma arquitetura (de software, e corporativa);
- Exemplos:
 - TOGAF;
 - DODAF;
 - Zachman;

Conceitos

- Visão e Pontos de Vista
- Decisão Arquitetural
- Padrão Arquitetural
- Concerns
- Atributos de Qualidade
- Linguagem de Descrição Arquitetural
- Framework Arquitetural
- **Reúso**

Conceitos

- Visão e Pontos de Vista
- Decisão Arquitetural
- Padrão Arquitetural
- Concerns
- Atributos de Qualidade
- Linguagem de Descrição Arquitetural
- Framework Arquitetural
- **Reúso**

Reúso

- A regra de ouro da Engenharia de Software para garantir produtividade e qualidade de software;
- Consiste no reaproveitamento de componentes, código, documentos, e demais artefatos de engenharia, para desenvolver um novo produto derivado de um anterior;
- Exemplos:
 - Linhas de Produto;
 - Desenvolvimento de Software Baseado em Componentes;

Reúso

- O uso de arquitetura de software já fomenta, se bem realizada, o reúso de software.

Introdução à Arquitetura de Software

Prof. Pedro Henrique Dias Valle

Adaptado de:
Profa. Elisa Yumi Nakagawa
Prof. Valdemar Neto