

Documentação de Arquiteturas de Software

Prof. Fernando Castor Filho

1

Recapitulando: Arquitetura de Software

- Projeto (*design*) de alto nível de um sistema de software
- Meio através do qual um sistema torna-se capaz de satisfazer diversos **requisitos**:
 - Segurança
 - Disponibilidade
 - Escalabilidade
 - Desempenho
- Normalmente representadas através de diversas **visões**
 - Cada uma enfatiza diferentes características do sistema

Como descrever essas visões, porém?

2

Documentação de Arquiteturas

- Documenta-se **visões**
 - Não se documenta a arquitetura “inteira”
- A escolha das visões a documentar depende
 - Da informação que se deseja documentar/ comunicar
 - Das análises que serão feitas
 - De quem usará a documentação
 - Do objetivo de se documentar
- Também é preciso analisar o grau de **detalhe** necessário
- A **experiência** do arquiteto é fundamental

3

Usos de Documentação Arquitetural

- Arquitetos e engenheiros de requisitos:
 - Negociar *trade-offs* entre requisitos conflitantes
- *Arquiteto e projetistas das partes*:
 - Resolver disputas por recursos e estudar desempenho esperado e outras características relativas ao consumo de recursos
- *Implementadores*:
 - Definir restrições que precisam ser respeitadas por atividades subsequentes de desenvolvimento
- *Testadores*:
 - Estabelecer o comportamento caixa-preta das partes que precisam trabalhar juntas
- *Desenvolvedores de outros sistemas*:
 - Definir interfaces providas e requeridas

4

Usos de Documentação Arquitetural

- Especialistas em qualidade:
 - Prover modelos analíticos que permitam a análise (semi-) automatizada da arquitetura
- *Gerentes*:
 - Quebrar a equipe de desenvolvimento em termos de módulos arquiteturais de alta granularidade
- *Responsáveis por manutenção*:
 - Entender o impacto de uma mudança
- *Gerentes de linhas de produtos*:
 - Determinar se um novo membro de uma família de produtos está ou não dentro do escopo da linha
- *Novo arquiteto*
 - Entender a arquitetura de um sistema pré-existente

5

Como Documentar Arquiteturas

- *Três passos simples*:
 1. Escolher as visões relevantes
 2. Documentar as visões escolhidas
 3. Documentar informação que se aplica a mais de uma visão

6

Tipos de Visão Arquitetural

- Em geral, visões descrevem um entre três aspectos complementares de um sistema
 - Como ele é dividido em **unidades de implementação**
 - Como ele é estruturado em unidades que têm **comportamento e interações** em tempo de execução
 - Como ele se relaciona a **elementos externos** que não sejam software

7

Tipos de Visão Arquitetural

- Em geral, visões descrevem um entre três aspectos complementares de um sistema
 - Como ele é quebrado em **unidades de implementação => MÓDULOS**
 - Como ele é estruturado em unidades que têm **comportamento e interações** em tempo de execução => **COMPONENTES E CONECTORES**
 - Como ele se relaciona a **elementos externos** que não sejam software => **ALOCUÇÃO**

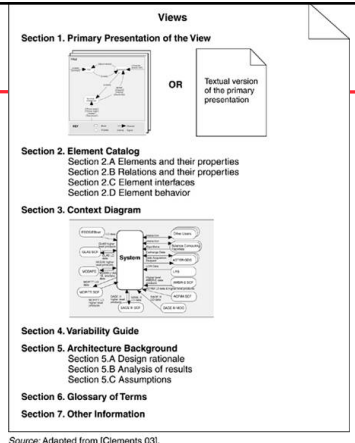
8

Exemplos de Visões

- Visões de Módulos:
 - Decomposição hierárquica
 - Classes
 - Camadas
- Componentes e conectores
 - Interação/Clientes e Servidores
 - Processos
- Alocação
 - Implantação/Física
 - Quebra de trabalho

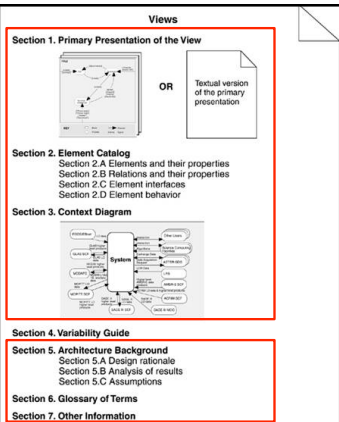
9

Documentando Visões



Documentando Visões

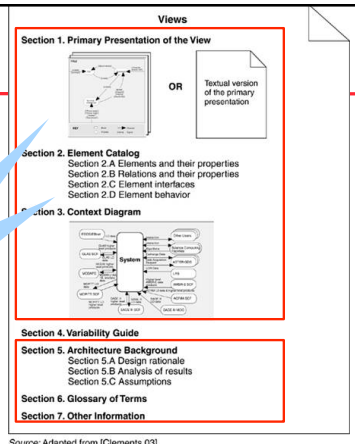
Seções relevantes para o projeto



Documentando Visões

Seções relevantes para o projeto

O conteúdo destas seções depende diretamente da visão documentada



Documentando Visões de Módulos

- **Camadas:**
 - Diagramas de **bloco**s (na UML => pacotes) mostrando os pontos de interação entre as camadas
 - Especialmente importante quando as camadas não são estritas
- **Classes**
 - Diagramas de classes da UML
- **Decomposição hierárquica**
 - Diagramas de pacotes da UML
 - Cada pacote representando um sub-sistema/componente arquitetural
 - Relacionando cada componente arquitetural com os sub-sistemas que o compõem
- **Documentação de interfaces**

13

Documentação de Comportamento

- **Sequências de mensagens na interação entre dois ou mais elementos em tempo de execução**
 - Diagramas de interação da UML: sequência e colaboração
- **Estados do sistema e de seus componentes**
 - **Statecharts**
 - Incluem as transições que levam de um estado para outro
 - Verificação automática
- **Documentação de interfaces**

14

Documentação de Interfaces

Section 2C. Element Interface Specification

- Section 2.C.1. Interface identity
- Section 2.C.2. Resources provided
 - Section 2.C.a. Resource syntax
 - Section 2.C.b. Resource semantics
 - Section 2.C.c. Resource usage restrictions
- Section 2.C.3. Locally defined data types
- Section 2.C.4. Exception definitions
- Section 2.C.5. Variability provided
- Section 2.C.6. Quality attribute characteristics
- Section 2.C.7. Element requirements
- Section 2.C.8. Rationale and design issues
- Section 2.C.9. Usage guide

Source: Adapted from [Clements 03].

15

Documentando Múltiplas Visões

Documentation across Views

How the document is organized:

- 1.1 View catalog
- 1.2 View template

What the architecture is:

- 2.1 System overview
- 2.2 Mapping between views
- 2.3 List of elements and where they appear
- 2.4 Project glossary

Why the architecture is the way it is:

- 3.1 Rationale

Source: Adapted from [Clements 03].

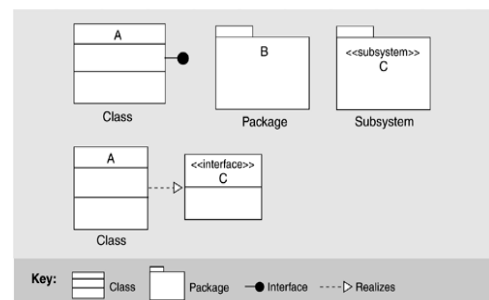
16

Escolhas de Projeto (Design Rationale)

- **Rationale** => justificativas para as escolhas
- **Mostram como a arquitetura inteira **soluciona** os problemas propostos pelos **requisitos****
- **Ajudam a **explicar** questões como as seguintes:**
 - Como o sistema satisfaz seus **requisitos**?
 - Como a arquitetura pode ser afetada pela adição de novas funcionalidades?
 - Que restrições são impostas sobre desenvolvedores implementando a solução?
 - Quais decisões alternativas foram rejeitadas?

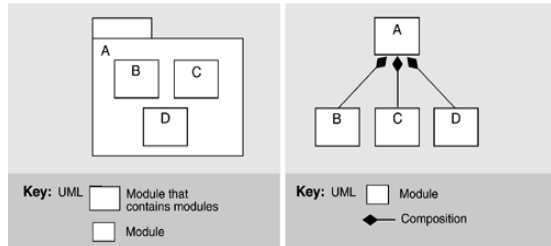
17

UML: Módulos



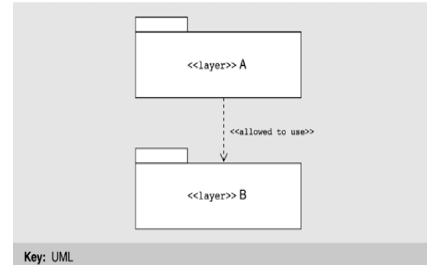
18

UML: Módulos



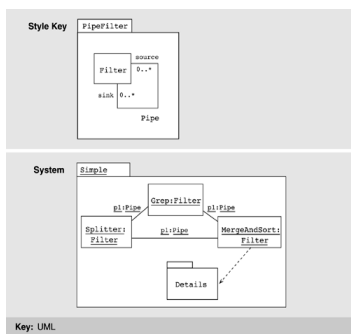
19

UML: Módulos



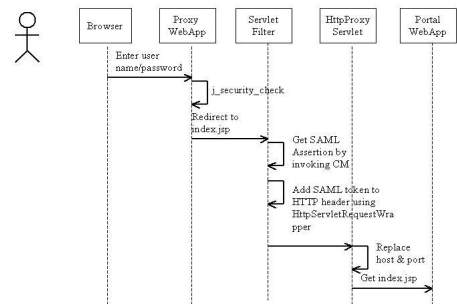
20

UML: Componentes e Conectores



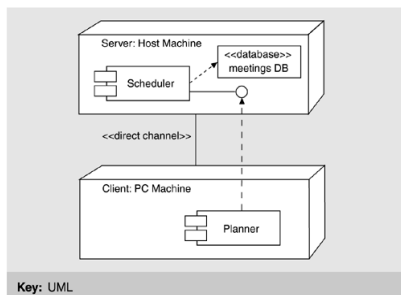
21

UML: Interação



22

UML: Alocação



23