SÃOJUDAS UNIVERSIDADE

Curso - Especialização em Engenharia de Software

Disciplina: Aspectos Avançados do Teste de Software: Revisões & Inspeções

Prof. Edson Saraiva

Revisão do Projeto

Agenda

- Métodos de avaliação da arquitetura
- Identificação de Requisitos não Funcionais - Atributos de Qualidade
- Documentação da Arquitetura
- Inspeção de documentos arquiteturais

2

Introdução

Projeto da Arquitetura

- O crescimento na utilização e a complexidade que envolve os sistemas de software aumentam a necessidade de se manter abordagens mais rigorosas para planejar as decisões adotadas nas fases iniciais do projeto

Requisitos implícitos

- Existe um conjunto de requisitos implícitos, frequentemente não mencionados (por exemplo o desejo de facilidade de uso e boa manutenibilidade) que quando não atendidos implicam em uma qualidade de software suspeita.

Requisitos não funcionais

- Estes requisitos frequentemente estão relacionados aos aspectos que envolvem decisões de projeto que vão definir a arquitetura do software.

3

Introdução

Projeto da Arquitetura

- Estas decisões irão afetar aspectos estruturais e comportamentais do sistema e são as mais difíceis de serem corrigidas quando não atendem as necessidades do cliente.

Arquitetura de Software

- Representa a estrutura, ou conjunto de estruturas, que compreende os elementos de software, suas propriedades externamente visíveis e seus relacionamentos (Bass, 2003).

4

Introdução

Arquitetura de Software

- Essa estrutura é utilizada como ferramenta para comunicar a solução projetada entre os diversos envolvidos que participam do processo de desenvolvimento do software (Clements, 2004).

Documento Arquitetural

- Para que essa comunicação seja possível, essa estrutura deve ser descrita através de uma representação, conhecida como documento arquitetural.

5

Introdução

Identificar os RNF

- Um dos desafios chave na produção de uma arquitetura de software de alta qualidade é identificar e entender os requisitos arquiteturalmente relevantes para o software.

Projeto da Arquitetura

- É o primeiro estagio no desenvolvimento de um software nos quais os requisitos de qualidade podem ser estabelecidos (BASS et al, 2003).

6

Métodos de Avaliação da Arquitetura

Revisão da Arquitetura

- A dificuldade de se corrigir decisões que estabelecem a arquitetura do software torna a atividade de revisão da arquitetura relevante para o sucesso do projeto e para melhoria da qualidade do software.

Abordagens

- Existem diversas abordagens descritas na literatura que objetivam avaliar a qualidade da arquitetura de software (Babar, 2004).

7

Métodos de Avaliação da Arquitetura

- Scenario based Architecture Analysis Method (SAAM)
- Architecture Tradeoff Analysis Method (ATAM)
- Active Reviews for Intermediate Design (ARID)
- SAAM for Evolution and Reusability (SAAMER)
- Architecture-Level Modifiability Analysis (ALMA)
- Architecture-Level Prediction of Software Maintenance
- (ALPSM)
- Scenario-Based Architecture Reengineering (SBAR)
- SAAM for Complex Scenarios (SAAMCS)
- SAAM in domain-Centric and Reuse-based development (ISAAMCR)

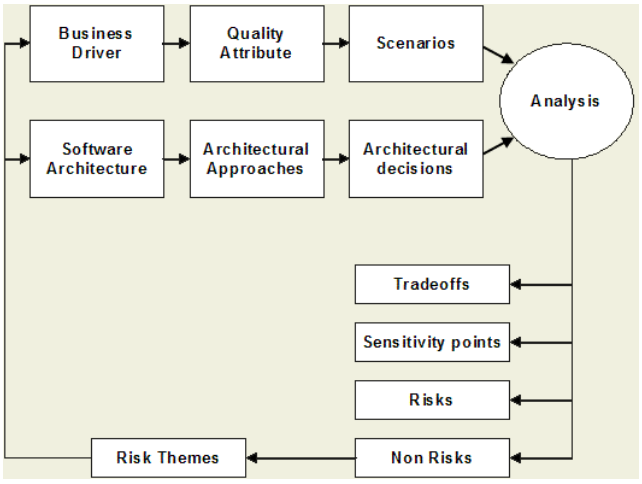
8

ATAM

Método de Avaliação	• Não existe consenso nos aspectos técnicos e não técnicos que um método deve atender e quais dos métodos existentes é mais indicado para um cenário específico.
ATAM	• Na literatura existem relatos de utilização com sucesso do método ATAM do SEI para avaliação da arquitetura de software.
Análise e Melhoria do Projeto Arquitetural	• O método é geralmente aplicado para a versão final de uma arquitetura de software, ele também é utilizado como um método de analise e melhoria do projeto arquitetural em processos de desenvolvimento de software baseados na arquitetura.a..

9

ATAM – Fluxo Conceitual



10

Avaliação da Arquitetura

- Envolve:
  1. Identificação dos requisitos não funcionais (cenários de qualidade)
  2. Seleção da abordagem arquitetural para atender aos requisitos não funcionais
  3. Documentação da arquitetura – visões arquiteturais
  4. Mecanismo de classificação de defeitos
  5. Estabelecer indicadores da qualidade arquitetural

11

Identificação dos RNF

- Uma aplicação é caracterizada por muitos atributos de qualidade.
- Atributos de qualidade são caracterizações especificas ou propriedades que podem ter algum valor quantitativo ou qualitativo e são mensuráveis.
- Um requisito de qualidade é uma especificação de um valor aceitável de um atributo de qualidade que deve estar presente em um sistema (ALBIN, 2003).

12

Identificação dos RNF

- Atributos de qualidade podem ser difíceis para especificar, implementar, medir e testar.
- Razões comuns das dificuldades envolvidas para se estabelecer aspectos de qualidade em um sistema incluem:
  - Desconhecimento da importância dos atributos de qualidade
  - Mecanismos inadequados para expressar e especificar requisitos de qualidade
  - Dificuldades para projetar o sistema de acordo com atributos de qualidade
  - Documentação de projeto inadequada
  - Falhas no controle da qualidade

13

Atributos de Qualidade

Atributos de qualidade

- Requisitos não funcionais podem ser caracterizados através de cenários que descrevem atributos de qualidade (BASS, 2003).

Cenário geral de qualidade

- O cenário geral fornece um guia para especificar uma classe de requisitos de qualidade

14

Cenário Geral

Fonte do estimulo	Entidade humana ou computacional que gera o estimulo.
Estimulo	Condição que precisa ser considerada quando chega ao sistema.
Ambiente	O estimulo ocorre dentro de certas condições, o sistema pode estar em uma condição de sobrecarga, ou estar executando quando o estimulo ocorre.
Artefato	O artefato que é estimulado, pode ser o sistema como um todo ou parte dele.
Resposta	A resposta é a atividade assumida depois da chegada do estimulo.
Medida da Resposta	Quando a resposta ocorre, ela deve ser mensuravel de forma que os requisitos possam ser testados.

15

Cenário Geral

- Um exemplo de um cenário geral para descrever requisitos de desempenho para um sistema seria:
- “Um evento periódico de uma fonte independente chega ao sistema sob condições normais. O sistema tem de processar o estímulo com certa latência.”

16

Cenário Geral

- O cenário geral deve ser refinado para um cenário concreto especifico do sistema que esta sendo analisado.
- “Um evento de um sensor X chega a cada 10 ms com o sistema operando sob condições normais. O sistema deve processar o estimulo em 1s.”

17

Cenário Geral

- Bass descreve cenários gerais para seis atributos importantes de qualidade:
  - ✓disponibilidade
  - ✓manutenibilidade
  - ✓desempenho
  - ✓segurança
  - ✓testabilidade
  - ✓usabilidade

18

Disponibilidade

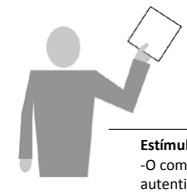
- Esta relacionada a falhas do sistema e as conseqüências associadas.

Fonte do estimulo	Indicações de falhas externas e internas do sistema.
Estimulo	Ocorrência de falha das seguintes classes: Omissão – um componente falha para responder a uma entrada. Trava – o componente repetidamente sofre falhas por omissão. Tempo – o componente responte de forma antecipada ou atrasada. Resposta – o componente responde com valores incorretos.
Artefato	Especifica os recursos necessarios para se obter alta disponibilidade tais como processador, canal de comunicação, processos ou armazenamento.
Ambiente	O estado do sistema quando a falha ocorre pode tambem afetar a resposta desejada do sistema.
Resposta	Existem muitas reaçoes possiveis para falha de um sistema. Entres elas pode-se incluir: registrar o evento (log), notificar usuários ou outros sistemas sobre a operação em modo degradado ou com menos capacidades ou com menos funções, desligamento de sistemas externos, ou tomar-se indisponível durante o reparo.
Medida da Resposta	A medida de resposta pode especificar uma porcentagem de disponibilidade, ou o tempo para reparo.

19

Disponibilidade

- Requisito de qualidade – o sistema deve registrar indisponibilidade do processo de autenticação de senhas do mainframe



Origem Externa:
-IBM RACF

Estímulo:
-O componente de autenticação não responde

Artefato:
disponibilidade do RACF

Ambiente:
-Normal
-Estimativa de 400 mil solicitações de autorização por hora

Resposta:
1. Sistema grava log de erro informando indisponibilidade do Senhas.
2. Sistema não autoriza credencial informada.

Medida de Resposta:
_Log de erro informando o momento em que ocorreu a indisponibilidade.
_O sistema fica Indisponível até que o RACF volte a operar

20

Exemplo

- REQ 15 – o sistema deve continuar operando mesmo em uma falha do sistema de RA com restrições somente para empréstimo
- Cenário geral

Fonte do estimulo	Indicações de falhas externas e internas do sistema.
Estimulo	Ocorrência de falha das seguintes classes: Omissão – um componente falha para responder a uma entrada. Trava – o componente repetidamente sofre falhas por omissão. Tempo – o componente responte de forma antecipada ou atrasada. Resposta – o componente responde com valores incorretos.
Artefato	Especifica os recursos necessarios para se obter alta disponibilidade tais como processador, canal de comunicação, processos ou armazenamento.
Ambiente	O estado do sistema quando a falha ocorre pode tambem afetar a resposta desejada do sistema.
Resposta	Existem muitas reaçoes possiveis para falha de um sistema. Entres elas pode-se incluir: registrar o evento (log), notificar usuários ou outros sistemas sobre a operação em modo degradado ou com menos capacidades ou com menos funções, desligamento de sistemas externos, ou tomar-se indisponível durante o reparo.
Medida da Resposta	A medida de resposta pode especificar uma porcentagem de disponibilidade, ou o tempo para reparo.

21

Exemplo

- Cenário específico

Fonte do estimulo	Uma falha externa do sistema de registro acadêmico.
Estimulo	Uma exception é disparada na tentativa de acessar o sistema de registro acadêmico
Artefato	Módulo de empréstimo
Ambiente	Operação normal
Resposta	Gravar o erro na tentativa de acesso e emitar uma mensagem para o usuário da indisponibilidade do sistema
Medida da Resposta	A consulta ao log de erros deve ser realizada em no máximo 3 minutos e a console do sistema deve informar o problema no acesso ao módulo de registro acadêmico.

22

Manutenibilidade

- A facilidade de modificação está relacionada ao custo da mudança.

Fonte do estímulo	Usuário final, desenvolvedor, administrador do sistema.
Estímulo	A parte especifica da mudança a ser feita. Uma mudança pode estar relacionada ao desejo para adicionar/excluir/ modificar/variar uma funcionalidade, atributo de qualiade, capacidade.
Artefato	A parte especifica que será mudada a funcionalidade do sistema, a plataforma, interface do usuário, ambiente ou outro sistema com o qual ele opera.
Ambiente	Especifica quando a mudança pode ser feita, em tempo de projeto, em tempo de compilação, em tempo de iniciação ou tempo de execução.
Resposta	Efetivar a mudança, testar e entregar.
Medida da Resposta	Tempo e custo financeiro, número de módulos afetados, tempo gasto na mudança.

23

Desempenho

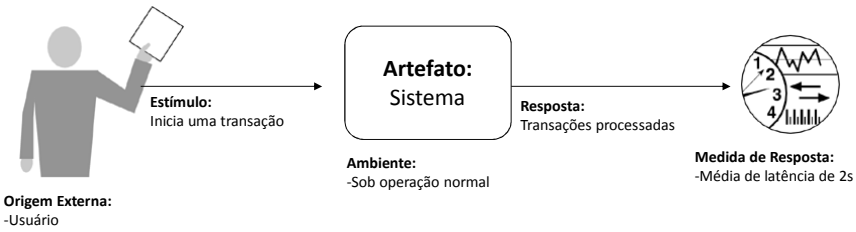
- O desempenho relaciona basicamente o tempo ocupado pelo sistema para responder a ocorrência de um evento.

Fonte do estímulo	O estímulo chega ou de fontes externas ou de fontes internas.
Estímulo	Chegada de eventos. A chegada pode ser categorizada por um padrão como periódico, esporadico, estocastico.
Artefato	O serviço do sistema.
Ambiente	Modo operacional tais como normal, emergencia ou sobrecarga.
Resposta	Processar o evento chegado.
Medida da Resposta	Latência ou limites definidos para o tempo de processamento, número de eventos que podem ser processados em um intervalo de tempo.

24

Desempenho

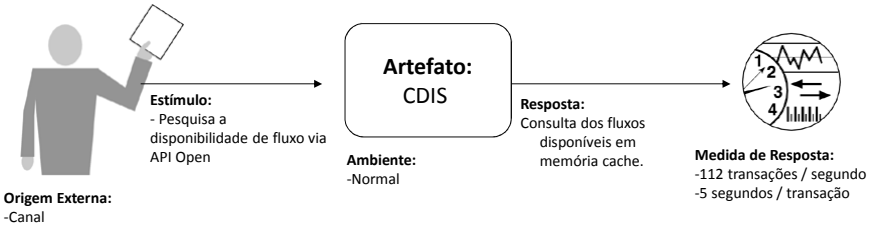
- Exemplo de um cenário geral de desempenho



25

Desempenho

- Requisito de qualidade – pesquisa via API Open



- Tática arquitetural - para garantir um alto throughput (transações/segundo), a consulta de fluxos utiliza tabela em memória cache.
- O desempenho pode ser incrementado com o acréscimo de células no cluster que são gerenciados pelo WebSphere.

26

Exemplo

- REQ31 - O sistema deve permitir o pagamento utilizando cartão de débito.
- Cenário geral de desempenho

Fonte do estímulo	O estímulo chega ou de fontes externas ou de fontes internas.
Estímulo	Chegada de eventos. A chegada pode ser categorizada por um padrão como periódico, esporadico, estocastico.
Artefato	O serviço do sistema.
Ambiente	Modo operacional tais como normal, emergencia ou sobrecarga.
Resposta	Processar o evento chegado.
Medida da Resposta	Latência ou limites definidos para o tempo de processamento, número de eventos que podem ser processados em um intervalo de tempo.

27

Exemplo

- Requisito não funcional de desempenho

Fonte do estímulo	Interna – solicitação de aprovação de uma transação originada em um ponto de venda
Estímulo	Solicitação de pagamento
Artefato	Autorizador
Ambiente	Operação normal
Resposta	O autorizador deve responder em no máximo 5 segundos
Medida da resposta	A transação é processada em no máximo 5 segundos

28

Segurança

- Avalia a habilidade do sistema para resistir a usos não autorizados (ataques) enquanto oferece serviços legítimos aos clientes.

Fonte do estímulo	A fonte do ataque pode ser humana ou outro sistema.
Estímulo	Ataque ou tentativa de quebrar a segurança.
Artefato	O alvo do ataque pode ser serviços mantidos pelo sistema ou dados.
Ambiente	O ataque vem de um sistema on-line ou offline, está conectado ou disconectado da rede.
Resposta	O sistema deve autorizar usuários legitimos e ao mesmo tempo rejeitar usuaários não autorizados, bloqueando o acesso e registrando a tentativa (log).
Medida da Resposta	Tempo e esforço e recursos requeridos de medidas de segurança para evitar ataques, probabilidade de detectar um ataque, probabilidade de identificar a responsabilidade individual para um ataque ou acesso/modificação de dados e ou servicos, meios de recuperação de dados/serviços.

29

Segurança

- Exemplo de um cenário geral de segurança

30

Testabilidade

- Refere-se a facilidade para se detectar falhas através de uma execução de teste típica.

Fonte do estimulo	Desenvolvedores de módulos (unidades), testadores de integração, testadores de sistema, ou pelo cliente. O projeto de teste pode ser realizado por outros desenvolvedores ou por um grupo externo.
Estimulo	Esta relacionado a um marco que o processo de desenvolvimento atende. Finalização da atividade de análise, ou um incremento da atividade de codificação tais como uma classe, a finalização da integração de um subsistema, ou a finalização do sistema.
Artefato	Um projeto, um pedaço do código ou o sistema todo é o artefato que está sendo testado.
Ambiente	O teste pode acontecer em tempo de projeto, em tempo de desenvolvimento, em tempo de compilação ou em tempo de entrega.
Resposta	Como a testabilidade está relacionada a observabilidade e a controlabilidade, a resposta desejada é que o sistema possa ser controlado para execução dos testes desejados e que a resposta para cada teste possa ser observada.
Medida da Resposta	As respostas são a porcentagem de comandos exercitados pelo teste, o tamanho da maior cadeia de tste (como uma medida de dificuldade para executar o teste) e estimativas de probabilidade de encontrar falhas adicionais.

31

Usabilidade

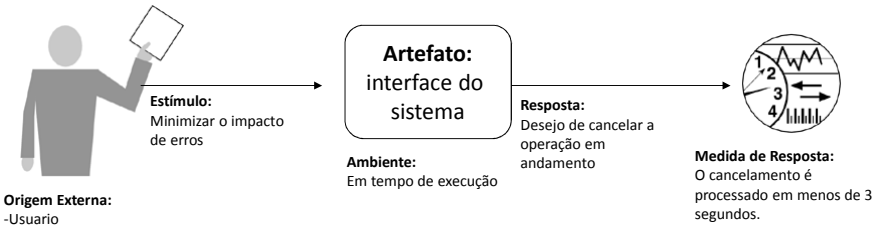
- Facilidades oferecidas para o usuário realizar uma tarefa desejada e o tipo de suporte ao usuário previsto pelo sistema.

Fonte do estimulo	O usuário final é sempre a fonte do estimulo.
Estimulo	Esta relacionado a um desejo do usuário final para usar o sistema eficientemente, aprender a usar o sistema, minimizar impactos de erros, adaptar o sistema para as necessidades do usuario, incrementar a confiança e satisfação.
Artefato	O sistema como um todo.
Ambiente	Em tempo de execução ou em tempo de configuração.
Resposta	O sistema deve prover o usuário com as caracteristicas de usabilidade necessarias ou antecipar as necessidades do usuário.
Medida da Resposta	Tempo e esforço para realizar uma tarefa, número de erros, satisfação do usuário, aprendizagem, taxa de operações realizadas com sucesso ou quantidade de tempo gasto quando um erro ocorre.

32

Usabilidade

- Exemplo de um cenário de usabilidade



33

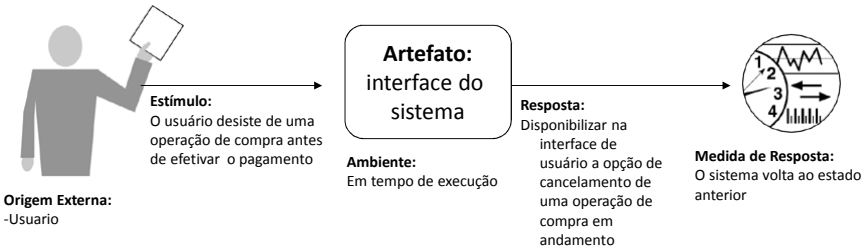
Usabilidade

- Facilidade de aprendizagem do funcionamento do sistema para usuários não experientes
- Uso do sistema eficientemente – facilidades que o sistema oferece para aumentar a eficiência do usuário.
- Minimizar o impacto de erros
- Adaptar o sistema as necessidades do usuário
- Incrementar a confiança e satisfação - feedbacks

34

Usabilidade

- Exemplo de um cenário de usabilidade



35

Análise Arquitetural

- O projeto de um sistema de software consiste de uma coleção de decisões que devem atender aos requisitos de qualidade especificados para o sistema.
- Identificar os cenários relevantes para arquitetura permite especificar os requisitos de qualidade para o sistema, mas não oferece ajuda para implementar a arquitetura considerando estes requisitos.

36

Análise Arquitetural

- Uma tática arquitetural é usada pelo arquiteto para atender um requisito de qualidade usando:
  - padrões de projeto
  - padrões arquiteturais ou estratégias arquiteturais.
- Cada tática especifica uma opção de projeto adotada para o sistema.

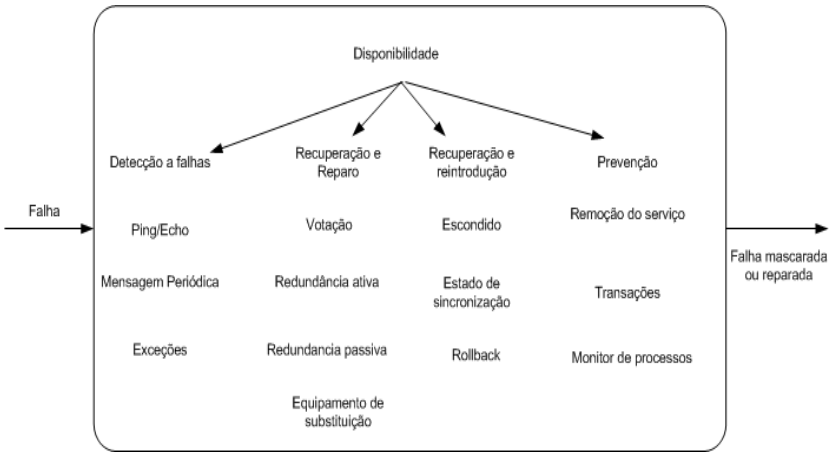
37

Tática Arquitetural - Disponibilidade

- Abordagens para manter disponibilidade de sistemas de software envolvem:
  - algum tipo de redundância
  - de monitoração para detectar a falha
  - algum tipo de recuperação quando a falha é detectada.

38

Tática Arquitetural - Disponibilidade



39

Tática Arquitetural - Disponibilidade

- Detecção de falhas:
  - Ping/echo – um componente emite um sinal e espera receber uma resposta (echo) dentro de um tempo predefinido.
  - Mensagens periódicas – exemplo uma máquina ATM pode enviar periodicamente o log das ultimas transações para um servidor se a mensagem falha o dispositivo de correção é acionado.
  - Exceções

40

Tática Arquitetural - Disponibilidade

- Recuperação e Reparo:
  - Votação
  - Redundância ativa
  - Redundância passiva
  - Substituição - um equipamento de substituição é configurado para substituir diferentes componentes que podem entrar em falha.

41

Tática Arquitetural-Disponibilidade

- Recuperação e reintrodução
  - Operação escondida: um componente que apresentou uma falha pode ser executado em modo oculto por um breve período de tempo para garantir que esta operando corretamente antes de restaurar o serviço
  - Sincronização de estado: quando é necessário restaurar um estado específico antes de entrar em operação
  - Checkpoint /rollback – o sistema é recuperado do último estado consistente.

42

Tática Arquitetural - Disponibilidade

- Prevenção de falhas:
  - Remoção preventiva
  - Transações manipulam vários passos seqüências para garantir atomicidade
  - Monitor de processos ao detectar uma falha exclui o processo e cria uma nova instancia do processo.

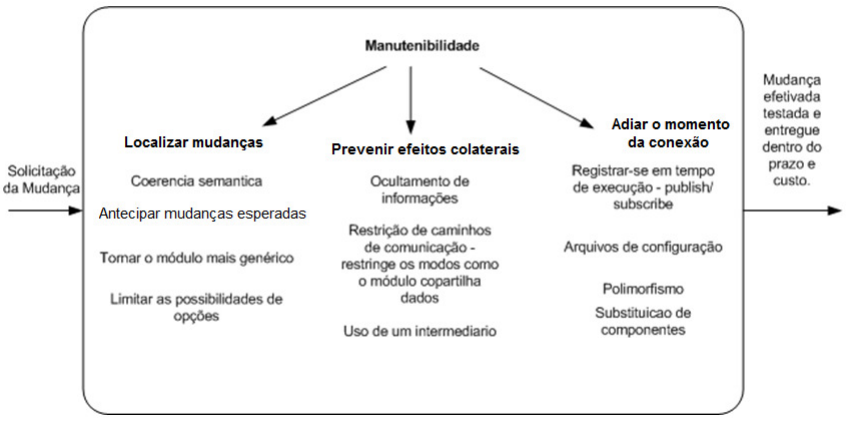
43

Tática Arquitetural - Manutenibilidade

- Táticas para planejar a manutenibilidade de um software têm como principal objetivo controlar o custo da mudança.
- Os cenários de manutenibilidade relacionam:
  - o que pode ser mudado,
  - quem faz a mudança
  - e quando a mudança é efetivada.

44

Tática Arquitetural-Manutenibilidade



45

Tática Arquitetural - Manutenibilidade

- Localizar mudanças
  - Manter coerência semântica – métricas de acoplamento e coesão
  - Antecipar as mudanças esperadas: planejar as modificações previstas com objetivo de minimizar o efeito das mudanças.
  - Generalizar o módulo
  - Limitar as opções possíveis de modificações

46

Tática Arquitetural - Manutenibilidade

- Prevenir efeitos colaterais
  - Ocultamento de informações (encapsulamento) análise dos vários tipos de dependências que um módulo pode ter com outro
- Adiar o momento da conexão
  - Registro em tempo de execução
  - Arquivo de configuração
  - Substituição de componente
  - Compatibilização com protocolo

47

Tática Arquitetural - Desempenho

- Táticas para planejar o desempenho do software são projetadas para gerar uma resposta a um evento que chega ao sistema dentro de algumas restrições de tempo.

48

Tática Arquitetural - Desempenho

- Demanda de recursos
  - Incremento da eficiência computacional
  - Redução da sobrecarga computacional
  - Gerência da taxa de eventos
  - Limita tempo de execução, tamanho de filas
- Gerenciamento dos recursos
  - Introduz concorrência
  - Manter múltiplas cópias
  - Aumentar a disponibilidade de recursos

49

Tática Arquitetural - Desempenho

- Arbitração de recursos
  - FIFO
  - Prioridade fixa

50

Documentação da Arquitetura

- A documentação da arquitetura deve servir para propósitos variados.
- Deve ser suficientemente abstrata para o entendimento de novos envolvidos.
- Deve conter informações suficientes para servir como base para atividades de análise.
- Deve ser suficientemente detalhada para servir como base da implementação.
- Prescritiva e descritiva (CLEMENTS, 2007)

51

Documentação da Arquitetura

- O processo de planejamento da documentação devem oferecer suporte as necessidades de revisão.
- Documentação para analises de desempenho pode ser bem diferente de uma documentação para dirigir a implementação

52

Documentação da Arquitetura

- O entendimento do uso da documentação da arquitetura é essencial para determinar a forma de documentação:
  - Educacional: envolvimento de novos membros na equipe
  - Como veículo primário para comunicação entre os envolvidos
  - Como base para análise do sistema

53

Documentação da Arquitetura

- Um dos conceitos mais importantes associados com a documentação da arquitetura de software é o de visão arquitetural.
- A arquitetura de software é uma entidade complexa que não pode ser descrita como uma única dimensão.
- A visão mais relevante depende dos objetivos de documentação(CLEMENTS, 2004).

54

Visões Arquiteturais

- No RUP a arquitetura é representada considerando um conjunto típico de visões, denominado "modelo de visão 4+1 (KRUTCHEN, 1995)
- Na essência, são fragmentos que ilustram os elementos "significativos em termos de arquitetura" dos modelos.

55

Visões Arquiteturais

- O modelo é composto das seguintes visões:
  - Visão de Casos de Uso – visão obrigatória.
  - Visão Lógica - quando tecnologia OO é utilizada esta relacionado ao modelo de objetos do projeto.
  - Visão de Processos - esta visão é opcional captura aspectos de concorrência e sincronização de threads, utilizada somente se o sistema tiver mais de um thread de controle e se os threads separados interagirem ou forem dependentes entre si.

56

Visões Arquiteturais

- O modelo é composto das seguintes visões:
  - Visão física – descreve o mapeamento do software no hardware e reflete o aspecto distribuído
  - Visa de desenvolvimento – descreve a organização estática do código no ambiente de desenvolvimento.

57

Processo de Projeto do Sistema

Projeto da arquitetura	• Os subsistemas e seus relacionamentos são identificados
Especificação dos subsistemas	• Uma especificação abstrata dos serviços e restrições de cada subsistema é produzida
Projeto de componentes	• Os serviços são alocados para diferentes componentes e suas interfaces são projetadas
Projeto das estruturas de dados	• As estruturas de dados usadas na implementação são especificadas e detalhadas
Projeto de algoritmos	• Os algoritmos são especificados e detalhados

58

Projeto da Arquitetura

Decomposição

- Sistemas complexos são sempre decompostos em subsistemas que fornecem algum conjunto de serviços relacionados.

Projeto da Arquitetura

- É um processo inicial de projeto, que consiste em identificar os subsistemas e estabelecer uma estrutura de controle e de comunicação entre os subsistemas.

59

Projeto da Arquitetura

Codifica/remenda

- Um caminho comum para muitos programadores é rapidamente estabelecer algum tipo de organização aleatória e não definir nenhuma regra ou protocolo para conduzir as interações entre módulos ou mesmo uma estratégia de agrupamento.

Comunicação entre os módulos de programação

- Desta forma os componentes podem falar entre si sem qualquer limitação ou protocolo

Aplicativos simples

- Este modelo é muito utilizado e relativamente eficiente para aplicativos que necessitam de mecanismos simples de entrada de dados e atualização em banco de dados

60

Projeto da Arquitetura

Manutenibilidade

- Na medida em que a complexidade da lógica do aplicativo aumenta com regras de negócios, validações e cálculos geralmente embutidos diretamente na interface de usuário tornam estas aplicações difíceis de manter.

Duplicação de código

- Outra consequência desta abordagem é a duplicação de código. Isto implica que uma simples mudança resulta em alterações em muitas interfaces de comunicação com o usuário que possuem códigos similares.

61

Projeto da Arquitetura

Processo de Projeto do Sistema

- Representa uma ligação crítica entre os processos de engenharia de projeto e o de requisitos.

Projeto da Arquitetura

- Identifica os principais componentes de um sistema e os mecanismos de comunicação entre eles.

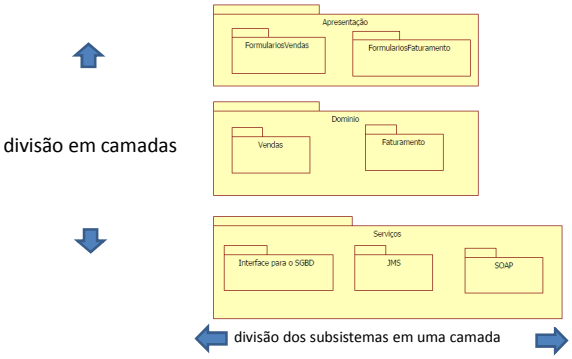
62

Projeto da Arquitetura

Projeto de subsistema

- É uma decomposição abstrata de um sistema em componentes de alta granularidade, cada um dos quais podendo ser um sistema substancial independente.

divisão em camadas



63

Projeto da Arquitetura

Protocolo de comunicação

- A colaboração pode ser planejada das camadas mais altas para as mais baixas
- Uma camada pode ser acessada por qualquer camada superior
- Uma camada somente é acessada pela camada imediatamente superior

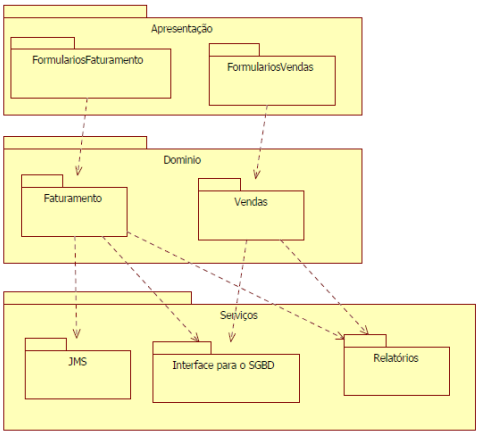
Visão lógica

- É interessante incluir um diagrama da visão lógica que mostre as dependências entre camadas e pacotes

64

Projeto da Arquitetura

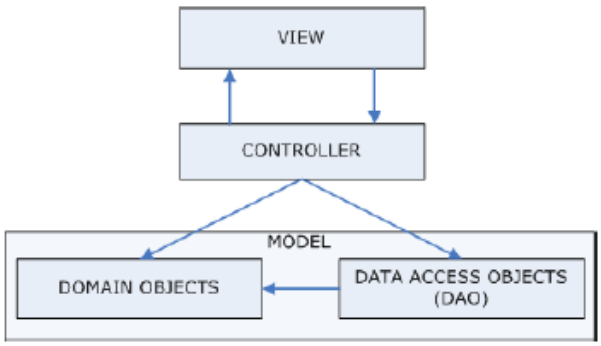
- Visão Lógica



65

Projeto da Arquitetura - MVC

- Objetos da camada de visão estão desacoplados de qualquer representação de estrutura de dados e objetos do modelo



66

Projeto de Arquitetura

Vantagens em projetar e documentar explicitamente	• Bass (2003) descreve três vantagens em projetar e documentar explicitamente uma arquitetura de software.
Comunicação	• Comunica a arquitetura para os envolvidos.
Análise do sistema	• Decisões de projetos de arquitetura tem profundo efeito sobre se o sistema pode atender aos requisitos críticos – desempenho, confiabilidade.
Reuso em larga escala	• A arquitetura do sistema é muitas vezes a mesma para requisitos similares.

67

Projeto de Arquitetura

Considera aspectos importantes logo no inicio	• Serve como um plano de projeto, ferramenta para gerenciar a complexidade, ocultar detalhes e permitir que os projetistas enfoquem nas abstrações principais do sistema.
Arquitetura afeta os atributos de qualidade	• A arquitetura de um sistema afeta o desempenho, facilidade de distribuição e de manutenção de um sistema (RNF).
Estilo arquitetural	• O estilo e a estrutura especifica escolhida para uma aplicação dependem dos requisitos não funcionais do sistema.
Conflitos	• Obviamente existem conflitos potenciais entre algumas dessas arquiteturas, o uso de componentes de alta granularidade aprimora o desempenho, o uso de componentes de baixa granularidade facilita a manutenção.
Solução	• Se os requisitos conflitantes forem requisitos importantes do sistema, em alguns casos, ambos podem ser atendidos pelo uso de estilos de arquitetura diferentes para diferentes partes do sistema.

68

Inspeção de Documentos Arquiteturais

- Identificação dos requisitos não funcionais
- Seleção da abordagem arquitetural
- Documentação da arquitetura – visões da arquitetura
- Avaliar a consistência da documentação – avaliam a consistência entre as visões estabelecidas para o projeto: ao analisar todos os diagramas foi identificado algum elemento arquitetural que não possua relacionamentos, ficando isolado dos demais.
- Avaliar o atendimento aos requisitos funcionais – permitem verificar se todas as funcionalidades especificadas foram atendidas pela arquitetura e se todos os elementos arquiteturais foram definidos com base nos requisitos especificados.

69

Referencias

- Bass, L., P. Clements and R. Kazman, Software Architecture in Practice, 2ed, Addison Wesley, 2003
- Clements, P., F. Bachmann, L. Bass, D. Garlan, J. Ivers, R. Little, R. Nord and J. Stafford, Documenting Software Architectures, Addison-Wesley, 2004
- Babar, M.A. et al., A Framework for Classifying and Comparing Software Architecture Evaluation Methods,Proceedings of the 2004 Australian Software Engineering Conference (ASWEC'04), 2004
- Albin,S. T., The Art of Software Architecture: Design Methods and Techniques, John Wiley & Sons, 2003
- Krutchen, F., Architectural Blueprints – The "4+1" View Model of Software Architecture, IEEE Software, 1995

70

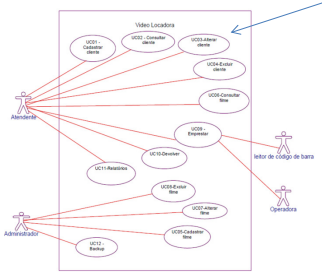
Atividade Prática

- Revisão de Requisitos - treinamento

2. Necessidades do Cliente

O sistema atual de gestão da locadora é ineficiente. O gerente deseja um sistema com objetivo de centralizar as informações, diminuir o tempo de espera dos clientes na loja, obter informações do faturamento por cliente, e permitir planejar uma campanha de marketing para capitalizar mais clientes e fidelizar os já existentes. É necessário que o sistema informe quais clientes são mais rentáveis e aqueles que dão o maior prejuízo. Com base nestas informações, serão planejados pacotes de locação com o intuito de aumentar o ticket médio de cada cliente (o quanto cada um contribui para a locadora).

1
2



1. Preparação - análise individual: cada individuo analisa o material independentemente
2. Reunião de inspeção – os defeitos encontrados individualmente são discutidos em equipe
3. Somente os defeitos comuns a equipe serão registrados no relatório

71