Aula 3

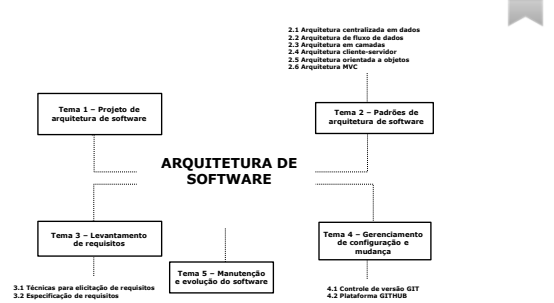
Engenharia de Software

Prof. Alex Mateus Porn

1

Conversa Inicial

2



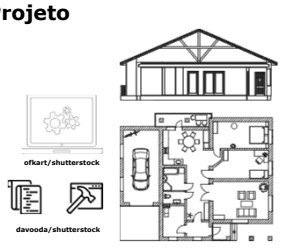
3

Projeto de arquitetura de software

4

Projeto

- **Para desenvolver um software, partimos do contexto geral desse sistema, para depois separarmos os processos em cada fase, de acordo com o modelo de processos (prescritivos ou ágeis) escolhido**



5

- **Projeto é:**
  - **"[...] um esforço temporário empreendido para criar um produto, serviço ou resultado único" (PMI, 2017)**
  - **"[...] um empreendimento planejado, orientado a resultados, possuindo atividades com início e término, para atingir um objetivo claro e definido" (Brasil, 2011)**

6

Projeto de software

- "[...] um processo em várias etapas em que as representações de dados e a estrutura do programa, as características de interfaces e os detalhes procedurais são sintetizados com base nos requisitos de informação" (Pressman, 2011, p. 229)

7

Projeto da arquitetura de software

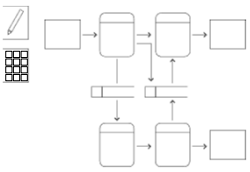
- "[...] representa a estrutura de dados e os componentes de programa necessários para construir um sistema computacional. Ele considera o estilo de arquitetura que o sistema assumirá, a estrutura e as propriedades dos componentes que constituem o sistema, bem como as inter-relações que ocorrem entre todos os componentes da arquitetura de um sistema" (Pressman, 2011, p. 229)

8

- Conforme Pressman (2011), as etapas de um projeto de arquitetura são:
  1. Projeto de dados
  2. Derivação da estrutura da arquitetura do sistema
  3. Análise de estilos ou padrões de arquitetura alternativos
  4. Implementação da arquitetura utilizando-se um modelo de processos

9

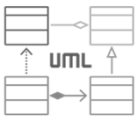
1. Projeto de dados
   - Levantamento de requisitos
   - Definição das entidades externas com as quais o software irá interagir
     - Diagrama entidade-relacionamento



Singkham/shutterstock

10

2. Derivação da estrutura da arquitetura do sistema
   - Identificação de arquétipos arquiteturais
   - Abstrações de elementos do sistema
   - Definição de classes e objetos
     - Linguagem UML
     - Diagrama de classes



laaisee/shutterstock

11

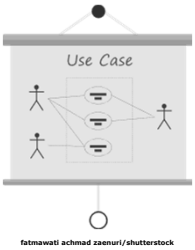
3. Análise de estilos ou padrões de arquitetura alternativos
   - Define-se o padrão de arquitetura de software a ser implementado
4. Implementação da arquitetura utilizando-se um modelo de processos
   - Implementação da arquitetura

12

- **Pfleeger (2004) destaca que o projeto é um processo iterativo composto por:**
  1. **Projeto conceitual**
  2. **Projeto técnico**

13

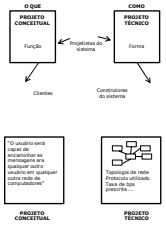
1. **Projeto conceitual**
   - **Apresenta ao cliente exatamente o que o sistema fará**
   - **Sendo aprovado, é traduzido, dando origem ao projeto técnico**
     - **Linguagem UML**
     - **Diagramas de casos de uso**



fatmawati achmad zaenuri/shutterstock

14

2. **Projeto técnico**
   - **Determinar hardware necessário**
   - **Determinar software necessário**
   - **Descrever a forma que o sistema terá**



Fonte: Pfleeger, 2004, p. 160-161

15

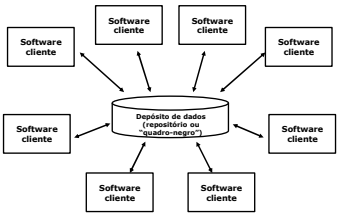
Padrões de arquitetura de software

16

- **"[...] é como uma descrição abstrata, estilizada, das práticas recomendadas que foram testadas e aprovadas em diferentes subsistemas e ambientes" (Sommerville, 2018)**

17

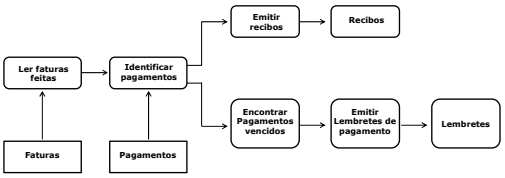
Arquitetura centralizada em dados



Fonte: Wazlawick, 2013, p. 62

18

Arquitetura de fluxo de dados

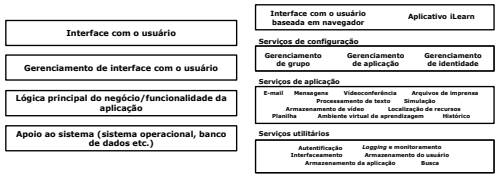


Fonte: Sommerville, 2018, p. 162

19

Arquitetura em camadas

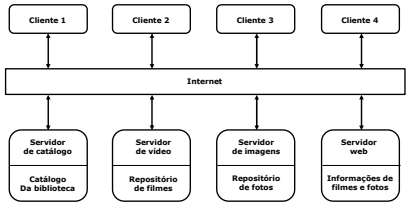
- **Sistema de ensino on-line**



Fonte: Sommerville, 2018, p. 155-156

20

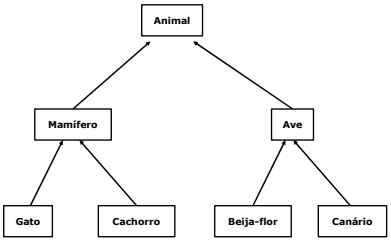
Arquitetura cliente-servidor



Fonte: Sommerville, 2018, p. 161

21

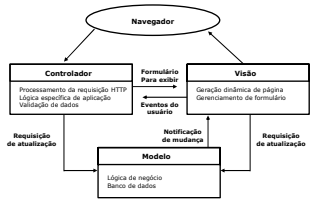
Arquitetura orientada a objetos



22

Arquitetura modelo, visão e controlador (MVC)

- **Aplicação web**



Fonte: Sommerville, 2018, p. 156

23

Levantamento de requisitos

24

Requisitos de software

- "[...] são as descrições do que o sistema deve fazer, os serviços que oferecem e as restrições a seu funcionamento. Esses requisitos refletem as necessidades dos clientes para um sistema que serve a uma finalidade determinada, como controlar um dispositivo, colocar um pedido ou encontrar informações" (Sommerville, 2018, p. 85)
- "[...] são classificados como requisitos funcionais e não funcionais" (Sommerville, 2018, p. 88-89)

25

Requisitos funcionais

- Declarações dos serviços que o sistema deve fornecer
- Como o sistema deve agir com relação a determinadas entradas
- Como o sistema deve se comportar em determinadas situações
- O que o sistema não deve fazer

26

Requisitos não funcionais

- Restrições sobre os serviços ou funções oferecidas pelo sistema
  - Restrições de tempo
  - Restrições sobre o processo de desenvolvimento
  - Restrições impostas por padrões
- Aplicam-se ao sistema como um todo, em vez de às características individuais ou aos serviços

27

Técnicas para elicitação de requisitos

- Entrevistas
  - Formais ou informais, com usuários e demais partes envolvidas no sistema
- Cenários
  - Podem ser escritos como texto, suplementados por diagramas, telas, entre outros

28

- Casos de uso
  - Abordagem mais estruturada de cenários. Identifica os atores envolvidos em uma iteração e dá nome ao tipo de iteração
- Etnografia
  - Técnica de observação que pode ser usada para compreender os processos operacionais e ajudar a extrair os requisitos de apoio para esses processos

29

Exemplo de cenário (Sommerville, 2011, p. 74)

- "Suposição inicial
  - O paciente é atendido em uma clínica médica por uma recepcionista. Ela gera um registro no sistema e coleta suas informações pessoais (nome, endereço, idade etc.). Uma enfermeira é conectada ao sistema e coleta o histórico médico do paciente

30

- **Normal**
  - **A enfermeira busca o paciente pelo sobrenome. Se houver mais de um paciente com o mesmo sobrenome, o nome e a data de nascimento são usados para identificar o paciente**
  - **A enfermeira escolhe a opção do menu para adicionar o histórico médico**

31

- **A enfermeira segue, então, uma série de *prompts* do sistema para inserir informações sobre consultas em outros locais, os problemas de saúde mental (entrada de texto livre), condições médicas (enfermeira seleciona condições do menu), medicação atual (selecionado no menu), alergias (texto livre) e informações da vida doméstica (formulário)**

32

- **O que pode dar errado**
  - **O prontuário do paciente não existe ou não pôde ser encontrado. A enfermeira deve criar um novo registro e registrar as informações pessoais**
  - **As condições do paciente ou a medicação em uso não estão inscritas no menu. A enfermeira deve escolher a opção "outros" e inserir texto livre com descrição da condição/medicação**

33

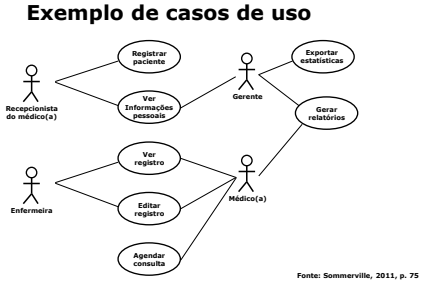
- **O paciente não pode/não fornecerá informações sobre seu histórico médico. A enfermeira deverá inserir um texto livre registrando a incapacidade/relutância do paciente em fornecer as informações. O sistema deve imprimir o formulário-padrão de exclusão afirmando que a falta de informação pode significar que o tratamento será limitado ou postergado. Este deverá ser assinado e entregue ao paciente**

34

- **Outras atividades**
  - **Enquanto a informação está sendo inserida, o registro pode ser consultado, mas não é editado por outros agentes**
- **Estado do sistema na conclusão**
  - **Usuário está conectado. O prontuário do paciente, incluindo seu histórico médico, é inserido no banco de dados e um registro adicionado ao log do sistema, mostrando o tempo de início e fim da sessão e a enfermeira envolvida" (Sommerville, 2011, p. 74)**

35

Exemplo de casos de uso



Fonte: Sommerville, 2011, p. 75

36

Elicitação de requisitos

- Sentenças em linguagem natural
  - Frases numeradas em linguagem natural
  - Cada frase deve expressar um requisito
- Linguagem natural estruturada
  - Linguagem natural em um formulário ou template
  - Cada campo fornece informações sobre um aspecto do requisito

37

- Notações gráficas
  - Modelos gráficos suplementados por anotações em texto
  - Diagramas de casos de uso e de sequência da UML
- Especificações matemáticas
  - Conceitos matemáticos como as máquinas de estados finitos ou conjuntos

38

Gerenciamento de configuração e mudança

39

Gestão de configuração e mudança

- "[...] é um conjunto de atividades destinadas a gerenciar as alterações identificando os artefatos que precisam ser alterados, estabelecendo relações entre eles, definindo mecanismos para gerenciar diferentes versões desses artefatos, controlando as alterações impostas e auditando e relatando alterações feitas" (Pressman, 2011, p. 514)

40

- Para Sommerville (2018), o gerenciamento de configuração envolve quatro atividades

1. Controle de versão
   - Controlar as várias versões dos componentes do sistema
   - Garantir que as mudanças feitas não interfiram nas outras
2. Construção de sistema
   - Reunir componentes, dados e bibliotecas do programa, compilando-os e ligando-os para criar um sistema executável

41

3. Gerenciamento de mudanças
   - Manter o controle das solicitações de mudança de clientes e desenvolvedores no software já entregue
   - Elaborar os custos e o impacto dessas mudanças, bem como decidir se e quando as alterações devem ser implementadas
4. Gerenciamento de lançamentos
   - Preparar o software para o lançamento externo e acompanhar as versões lançadas para uso do cliente

42

Sistema de controle de versões: Git

- Sistema de controle de versão de arquivos
- Possibilita o desenvolvimento de projetos
- Diversos desenvolvedores podem contribuir simultaneamente no mesmo projeto
- Permite que arquivos possam existir sem o risco de suas alterações serem sobrescritas

Postmodern Studio/Shutterstock

43

Configurando o Git

- Instalação
  - sudo apt-get install git
- Configurações
  - git config --global user.name "Alex Mateus"
  - git config --global user.email "alex@mateus.com.br"
  - git config --global color.ui true

Postmodern Studio/Shutterstock

44

Gerenciando o Git

- Criar o repositório Git
  - git init
- Criando o versionamento do projeto
  - git status
  - git add arquivo
  - git add .
  - git commit
  - git commit –m "descrição"
  - git commit –a –m "descrição"

Postmodern Studio/Shutterstock

45

- Visualizar logs do Git
  - git log
  - git log –p
  - git log –p -2
  - git log –stat
  - git log --pretty=online
  - git log --pretty=format "%h - %a, %ar : %s"
  - git log --since=2.days

Postmodern Studio/Shutterstock

46

- Remover arquivos prontos para *commit*
  - arquivo .gitignore
- Retirar arquivos do *add*
  - git reset HEAD arquivo
- Retornar os *commits* realizados
  - git reset HEAD~1
  - git reset HEAD~1 --soft
  - git reset HEAD~1 --hard

Postmodern Studio/Shutterstock

47

Manutenção e evolução de software

48

- Wazlawick (2013, p. 318) aponta que "uma vez desenvolvido, um software terá um valor necessariamente decrescente com o passar do tempo"
  - Falhas são descobertas
  - Requisitos mudam
  - Produtos menos complexos, mais eficientes ou tecnologicamente mais avançados são disponibilizados

49

- Para Pressman (2011, p. 662), "independentemente do domínio de aplicação, tamanho ou complexidade, o software continuará a evoluir com o tempo"
- São corrigidos erros
  - Quando há adaptação a um novo ambiente
  - Quando o cliente solicita novas características ou funções
  - Quando a aplicação passa por um processo de reengenharia

50

- Sommerville (2018, p. 246) enfatiza que "a manutenção de software é o processo geral de mudança depois que ele é liberado para uso, existindo três diferentes tipos de manutenção"
  - Correção de defeitos
  - Adaptação ambiental
  - Adição de funcionalidade

51

Referências

52

- PMI. Guia PMBOK: um guia do conhecimento em gerenciamento de projetos. 6. ed. EUA: Project Management Institute, 2017.
- BRASIL. Metodologia de Gerenciamento de Projetos do SISP. Brasília: Ministério do Planejamento, Orçamento e Gestão. Secretaria de Logística e Tecnologia da Informação, 2011.
- PFLEEGER, S. L. Engenharia de software: teoria e prática. 2 ed. São Paulo: Prentice Hall, 2004.

53

- WAZLAWICK, R. S. Engenharia de software: conceitos e práticas. São Paulo: Elsevier, 2013.
- SOMMERVILLE, I. Engenharia de software. 10 ed. São Paulo: Pearson Education do Brasil, 2018.
- _____. Engenharia de software. 9. ed. São Paulo: Pearson, 2011.
- PRESSMAN, R. S. Engenharia de software: uma abordagem profissional. 7. ed. Porto Alegre: AMGH, 2011.

54