

Engenharia de requisitos: é o processo que define as funcionalidades e restrições.

Requisitos funcionais: é a descrição de uma funcionalidade, de forma a detalhar o que o sistema deve fazer com entrada, ou em cenários específicos.

Não funcionais: são restrições, com tempo, padrões, confiabilidade.

- ↳ se não atendidos pode m inutilizar o software.
- ↳ Afetam a arquitetura do software
- ex: um requisito de proteção pode gerar requisitos não funcionais.

Pg 111,

requisitos estabelecem o que o software deve fazer bem com restrições sobre funcionamento e implementação.

Documentos de requisitos: é a declaração oficial dos requisitos.

Utilizados por clientes, gerentes, desenvolvedores, testes, manutenção

• Especificação: é a descrição de requisitos de forma completa, técnica, compreensível.

Engenharia de requisitos. Pg 19.

- ✓ Elaboração e análise de requisitos
 - des coberta de requisitos, stakeholders, custos de uso (criação de canários para identificar erros e Entrega).
- ✓ Validação de requisitos.
 - Objetivo é validar se os requisitos definem o software que o cliente espera, p/ os custos com erros de requisitos são altos
 - técnicas: revisões manuais, prototipagem, casos de teste
- ✓ Gerenciador de requisitos.
 - É o processo para lidar com os requisitos em constante mudança
 - novas hardware, prioridades do cliente mudam...

Modelagem de Sistemas

Apresenta modelos de visões associadas ao software, auxilia a entender os funcionalidades.

Models gráficos - UML

- forma de documentar um software

Diagramas:

- los de uso; - sequência; classe; estado, intervidades.

- ↳ Criam diferentes perspectivas para modelar um software.
- ↳ Modelos de conteúdo (externo ao software)
- ↳ Models de interação (los de uso e sequência)
- ↳ Models estruturais: criados durante o projeto de arquitetura, representam a organização do software em:
 - diagrama de classes

Pg 19-20

- ↳ Models comportamentais: comportamentos do software durante execução, a partir de estímulos.

Modelos de máquinas de estados: descreve comportamento do software em resposta a eventos externos e internos.

- diagrama de estados.

Projeto de Arquitetura identifica os elementos de um software e suas comunicações.

Usos:

- com stakeholders.
- Análise de requisitos não funcionais
- reuso em largo escala.

Características:

- Desempenho, proteção, segurança, disponibilidade, manutenibilidade

Representações da arquitetura.

- ↳ Diagrama de blocos
- ↳ modelos de arquitetura
 - visão de alto nível
 - facilita interações
 - documenta a arquitetura.
- ↳ Pg 12, decisões de projeto.

Reuso

Ferramenta para geração de softwares de mesmo domínio.

Linha de produto: tem arquitetura central, com variações parciais ao cliente.

Arquitetura pode ser projetada em torno de t de um padrão, assim expandindo o modelo e possibilidade utilização em outros situações.

Visões (4+1)

Visões do software, pg 15.

- lógica, desenvolvimento, processo, física.

+ 1 visão de uso.

Padrões de arquitetura

meio de representar, organizar, reutilizar e conhecer.

→ Boas práticas de projeto, que já foram experimentadas e testadas.

Ex:

MVC - Modelo Visão Controlador.

- Usado quando existem muitos meios de visualizar e interagir com dados.

Pg 12

→ Pedir explicação pg 19

Camadas.

modela interfaces dos elementos do software.

→ Organiza o software em conjunto de camadas, e cada uma contém nº de elementos, que têm funcionalidades.

Pg 21.

Usado quando adicionar novos recursos a sistemas existentes.

Repositório

- Elementos do software devem ter os dados
Usado quando o grande volume de dados é
compartilhado.

→ Todos os dados de um sistema são gerenciados em um repositório central.

Pg 4.

Cliente - Servidor.

funcionalidades são organizadas por serviços
e cada um tem seu servidor.

Usado quando os dados em bloco precisam
ser acessados a partir de série de locais.

Pg 4

Duto e Filtro

Cada componente (filtro) seja discreto e realize um tipo de transformação de dados,
e os dados fluem (como um duto) de um
componente para outro.

Usado em aplicações de processamento de
dados.

Aplicações

São softwares projetados para atender
diferentes necessidades, e são similares.

- Organiza o desenvolvimento

- Pontos de partida para o projeto de arquitetura.

Pg 14

Arquitetura de Sistemas Informáticos

- ↳ Organizados em arquitetura em camadas, baseadas em troncos (banco de dados)
- ↳ Em geral são webs.

Pg 23

Aula 04 - Dá exemplos dos padrões de projeto.

Projeto e Implementação

COTS - commercial Off the Shelf (Software de plataforma)

- ↳ Reuso: reaproveitamento de artefatos Níveis de reuso.

- abstracções
- componentes
- objeto
- sistemas

Pg 6

Padrões de Projeto

fazem o reuso e conhecimento.

- ↳ Consideram orientação a objeto.

Pg 8!

Pg 14?

Open Source

Abordagem de desenvolvimento, na qual o código fonte é publicado e voluntários participam do desenvolvimento.

Linux, PostgreSQL.

Reuso

- ↳ Objetos e funções.
- ↳ Componentes.
- ↳ Sistemas todo

Benefícios pg 5
Problemas pg 6

Panorama de reuso, pg 8
(multiplos técnicas de reuso)

Frameworks

entidades que podem ser reutilizadas,
só projeto de subsistema (elementos e instâncias de classes abstratas)

Ex

Framework web, feito por mvc
WAFs ↴

- Proteção , - Banco de dados
- Dinâmico, - interativas ; sessões

↳ Frameworks são arquitetura - esqueleto, fornecem classes abstratas e métodos, eventos.

Pg 18 e 21

Linha de Produtos de Software

Aplicações genéricas que são adaptadas para contexto específico.

Adaptação

- Adição de elementos
- Seleção de elementos por uma biblioteca
- modificações de elementos existentes

↳ Frameworks dão suporte técnico, já a LPS tem informações do domínio.

Especialização:

- de plataforma (desktop, mobile...)
- Processo

Pg 6! Arquitetura de LPS deve se dividir em subsistemas (camadas)

- Pg 7 e 8. {
1º Elabora requisitos
2º Encontra opções similares
3º Organizar os requisitos
4º Adoptar o software.

↳ COTS, software que permite a adaptação sem alterar código fonte.
mecanismos

internos de configuração, Pg 18

ERP: sistema genérico que oferece suporte a negócios, como compra e faturamento