TI MAPEADA · 300 MAPAS MENTAIS DE TI

Módulo 01

Engenharia de Software

31 mapas · mapas 001 a 031



Neste módulo:

001 Ciclo de Vida do Software
002 Modelo Cascata
003 Modelo Incremental
004 Modelo Espiral
005 Prototipação
006 RUP (Rational Unified Process)
007 Manifesto Ágil
008 Scrum: Papéis
009 Scrum: Eventos
010 Scrum: Artefatos
011 Kanban
012 XP (Extreme Programming)
013 Lean e FDD
014 Engenharia de Requisitos
015 Requisitos Funcionais e Não Funcionais
016 Casos de Uso
017 UML: Diagramas Estruturais
018 UML: Diagramas Comportamentais
019 Diagrama de Classes
020 Níveis de Teste
021 Caixa-Preta e Caixa-Branca
022 TDD e BDD
023 Qualidade: ISO/IEC 25010
024 CMMI
025 MPS.BR
026 Análise de Pontos de Função
027 Estimativas e Métricas
028 Gerência de Configuração
029 Manutenção de Software
030 Arquitetura de Software
031 Engenharia Reversa e Reengenharia

Ciclo de Vida do Software

O QUE É

Conjunto de fases que vai da concepção à retirada de um software.

1 Fases

- requisitos
- projeto
- implementação
- testes
- implantação
- manutenção

2 Modelos prescritivos

- cascata
- incremental
- espiral
- RUP

3 Modelos ágeis

- Scrum
- XP
- Kanban
- FDD

4 Escolha do modelo

- estabilidade dos requisitos
- riscos
- tamanho da equipe

5 Manutenção

- costuma ser a fase mais longa e cara

⚠ CAI EM PROVA

A manutenção consome a maior parte do custo total do ciclo de vida.

Modelo Cascata

O QUE É

Modelo sequencial: cada fase começa só quando a anterior termina.

1 Sequência

- requisitos
- projeto
- codificação
- testes
- manutenção

2 Vantagens

- simples de gerenciar
- documentação forte
- marcos claros

3 Desvantagens

- pouca flexibilidade
- erro descoberto tarde
- cliente vê o produto no fim

4 Quando usar

- requisitos estáveis e bem entendidos

5 Variação

- modelo em V liga cada fase a um nível de teste

⚠ CAI EM PROVA

Royce já admitia retornos entre fases; a versão rígida é uma simplificação.

Modelo Incremental

O QUE É

Entrega o sistema em partes funcionais, cada incremento soma funcionalidades.

1 Ideia

- núcleo primeiro
- incrementos sucessivos
- cada um é utilizável

2 Vantagens

- entrega valor cedo
- feedback do usuário
- risco menor

3 Desvantagens

- exige boa arquitetura inicial
- integração contínua

4 Diferença do iterativo

- incremental soma partes
- iterativo refina o todo

5 Uso

- prazos curtos com prioridades claras

⚠ CAI EM PROVA

Incremental e iterativo não são sinônimos, mas podem ser combinados.

Modelo Espiral

O QUE É

Modelo evolutivo de Boehm guiado pela análise de riscos a cada volta.

1 Quadrantes

- objetivos
- análise de riscos
- desenvolvimento
- planejamento

2 Foco

- riscos analisados em todo ciclo

3 Vantagens

- projetos grandes e críticos
- usa protótipos

4 Desvantagens

- caro
- exige especialistas em risco

5 Cada volta

- gera uma versão mais completa do sistema

⚠ CAI EM PROVA

A marca do espiral é a análise de riscos explícita em cada ciclo.

Material de estudo e revisão · uso pessoal

Prototipação

O QUE É

Construção rápida de versões preliminares para validar requisitos com o usuário.

1 Tipos

- descartável
- evolutivo

2 Fidelidade

- baixa (papel, wireframe)
- alta (interativo)

3 Vantagens

- esclarece requisitos
- reduz mal-entendidos

4 Riscos

- cliente acha que está pronto
- código improvisado vira produto

5 Uso

- requisitos vagos ou interface crítica

⚠ CAI EM PROVA

No protótipo descartável, o código não é reaproveitado no produto final.

RUP (Rational Unified Process)

O QUE É

Processo iterativo, guiado por casos de uso e centrado na arquitetura.

1 Fases

- concepção
- elaboração
- construção
- transição

2 Marcos

- objetivos
- arquitetura
- capacidade operacional
- release

3 Disciplinas

- requisitos
- análise e design
- implementação
- teste

4 Características

- iterativo e incremental
- usa UML
- gestão de riscos

5 Elaboração

- define e valida a arquitetura base

⚠ CAI EM PROVA

As fases do RUP são sequenciais, mas dentro delas há iterações.

Manifesto Ágil

O QUE É

Declaração de 2001 com 4 valores e 12 princípios para desenvolver software.

1 Valor 1

- indivíduos e interações mais que processos e ferramentas

2 Valor 2

- software funcionando mais que documentação abrangente

3 Valor 3

- colaboração com o cliente mais que negociação de contratos

4 Valor 4

- responder a mudanças mais que seguir um plano

5 Princípios

- entregas frequentes
- ritmo sustentável
- simplicidade

⚠ CAI EM PROVA

O manifesto não elimina documentação: só a valoriza menos que software funcionando.

Scrum: Papéis

O QUE É

O Scrum Team tem três responsabilidades e nenhuma hierarquia interna.

1 Product Owner

- maximiza valor
- gerencia o Product Backlog

2 Scrum Master

- líder que serve
- remove impedimentos
- ensina Scrum

3 Developers

- criam o incremento
- montam o Sprint Backlog

4 Time

- auto-gerenciável
- multifuncional
- até 10 pessoas

5 Guia 2020

- fala em accountabilities e não em papéis

⚠ CAI EM PROVA

O PO é uma pessoa só, não um comitê.

Material de estudo e revisão · uso pessoal

Scrum: Eventos

O QUE É

Eventos com duração fixa (timebox) que dão ritmo ao trabalho.

1 Sprint

- até 1 mês
- contém todos os eventos

2 Sprint Planning

- até 8 h
- por quê, o quê e como

3 Daily Scrum

- 15 min
- para os Developers

4 Sprint Review

- até 4 h
- inspeciona o incremento

5 Retrospectiva

- até 3 h
- melhora o processo

⚠ CAI EM PROVA

Os tempos máximos valem para Sprints de 1 mês; Sprints menores têm eventos menores.

Material de estudo e revisão · uso pessoal

Scrum: Artefatos

O QUE É

Representam trabalho ou valor; cada um tem um compromisso associado.

1 Product Backlog

- lista ordenada
- compromisso: Meta do Produto

2 Sprint Backlog

- meta + itens + plano
- compromisso: Meta da Sprint

3 Incremento

- degrau rumo à meta
- compromisso: Definição de Pronto

4 Definição de Pronto

- padrão de qualidade do incremento

5 Refinamento

- atividade contínua, não é evento formal

⚠ CAI EM PROVA

Burndown não é artefato oficial do Scrum.

Material de estudo e revisão · uso pessoal

Kanban

O QUE É

Método de gestão do fluxo de trabalho com quadro visual e limite de WIP.

1 Práticas

- visualizar
- limitar WIP
- gerenciar fluxo
- políticas explícitas

2 Quadro

- colunas de etapas
- cartões de tarefas

3 Métricas

- lead time
- cycle time
- throughput

4 Sistema puxado

- só puxa nova tarefa quando há capacidade

5 Diferença do Scrum

- sem sprints nem papéis obrigatórios

⚠ CAI EM PROVA

Limitar o WIP é o que diferencia Kanban de um simples quadro de tarefas.

Material de estudo e revisão · uso pessoal

XP (Extreme Programming)

O QUE É

Metodologia ágil focada em práticas de engenharia e qualidade técnica.

1 Valores

- comunicação
- simplicidade
- feedback
- coragem
- respeito

2 Práticas

- programação em pares
- TDD
- integração contínua

3 Mais práticas

- refatoração
- pequenos releases
- propriedade coletiva

4 Planejamento

- jogo do planejamento com histórias de usuário

5 Ritmo

- semana de 40 horas, ritmo sustentável

⚠ CAI EM PROVA

Programação em pares e TDD são práticas típicas do XP.

Material de estudo e revisão · uso pessoal

Lean e FDD

O QUE É

Lean elimina desperdícios; FDD organiza o desenvolvimento por funcionalidades.

1 Lean princípios

- eliminar desperdício
- amplificar aprendizado
- entregar rápido

2 Desperdícios

- trabalho parcial
- funcionalidade extra
- espera
- defeitos

3 FDD processos

- modelo geral
- lista de features
- planejar
- projetar
- construir

4 FDD feature

- pequena, entregue em até 2 semanas

5 FDD papéis

- arquiteto-chefe
- programador-chefe
- dono de classe

⚠ CAI EM PROVA

O FDD tem 5 processos, os 3 primeiros de concepção e os 2 últimos de construção.

Engenharia de Requisitos

O QUE É

Processo de descobrir, documentar, validar e gerenciar requisitos.

1 Atividades

- elicitação
- análise
- especificação
- validação
- gestão

2 Técnicas

- entrevista
- questionário
- workshop
- observação
- JAD

3 Problemas

- ambiguidade
- requisitos conflitantes
- mudança constante

4 Validação

- revisões
- protótipos
- geração de casos de teste

5 Rastreabilidade

- liga requisito à origem e à implementação

⚠ CAI EM PROVA

Validação verifica se é o sistema certo; verificação, se está certo.

Material de estudo e revisão · uso pessoal

Requisitos Funcionais e Não Funcionais

O QUE É

Funcionais dizem o que o sistema faz; não funcionais, como e com que qualidade.

1 Funcionais

- serviços
- reações a entradas
- comportamento

2 Não funcionais

- desempenho
- segurança
- usabilidade
- disponibilidade

3 Categorias de RNF

- produto
- organizacionais
- externos

4 Requisitos de domínio

- vêm da área de aplicação

5 Escrita

- verificável e mensurável

⚠ CAI EM PROVA

Tempo de resposta máximo é requisito não funcional.

Material de estudo e revisão · uso pessoal

Casos de Uso

O QUE É

Descrevem interações entre atores e o sistema para atingir um objetivo.

1 Elementos

- ator
- caso de uso
- fronteira do sistema

2 Include

- comportamento obrigatório reutilizado

3 Extend

- comportamento opcional ou condicional

4 Generalização

- herança entre atores ou casos de uso

5 Especificação

- fluxo principal
- alternativos
- exceções

⚠ CAI EM PROVA

No extend a seta parte do caso estendido para o base; no include, do base para o incluído.

Material de estudo e revisão · uso pessoal

UML: Diagramas Estruturais

O QUE É

Mostram a estrutura estática do sistema: o que existe e como se relaciona.

1 Classes

- classes, atributos, métodos e relações

2 Objetos

- instâncias em um momento

3 Componentes

- partes substituíveis do software

4 Implantação

- nós físicos e artefatos

5 Outros

- pacotes
- estrutura composta
- perfil

⚠ CAI EM PROVA

A UML 2.x tem 14 diagramas: 7 estruturais e 7 comportamentais.

UML: Diagramas Comportamentais

O QUE É

Mostram o comportamento dinâmico do sistema ao longo do tempo.

1 Casos de uso

- funcionalidades e atores

2 Atividades

- fluxo de ações, parecido com fluxograma

3 Máquina de estados

- estados de um objeto

4 Interação

- sequência
- comunicação
- tempo
- visão geral

5 Sequência

- troca de mensagens em ordem temporal

⚠ CAI EM PROVA

Diagramas de interação são um subgrupo dos comportamentais.

Diagrama de Classes

O QUE É

Representa classes, atributos, operações e relacionamentos.

1 Associação

- ligação estrutural com multiplicidade

2 Agregação

- losango vazio
- parte existe sem o todo

3 Composição

- losango cheio
- parte morre com o todo

4 Generalização

- triângulo vazio
- herança

5 Visibilidade

- + público
- - privado
- # protegido
- ~ pacote

⚠ CAI EM PROVA

Composição é mais forte que agregação: o ciclo de vida da parte depende do todo.

Níveis de Teste

O QUE É

Etapas de teste conforme a abrangência do que está sendo testado.

1 Unidade

- menor parte testável
- feito pelo desenvolvedor

2 Integração

- interfaces entre módulos

3 Sistema

- sistema completo contra requisitos

4 Aceitação

- usuário valida
- alfa e beta

5 Regressão

- garante que mudanças não quebraram o que funcionava

⚠ CAI EM PROVA

Regressão é um tipo de teste, não um nível.

Material de estudo e revisão · uso pessoal

Caixa-Preta e Caixa-Branca

O QUE É

Técnicas de teste com ou sem conhecimento da estrutura interna do código.

1 Caixa-preta

- baseada na especificação
- entradas e saídas

2 Técnicas caixa-preta

- partição de equivalência
- valor limite
- tabela de decisão

3 Caixa-branca

- baseada no código
- estrutura interna

4 Técnicas caixa-branca

- cobertura de comandos
- desvios
- caminhos

5 Caixa-cinza

- mistura das duas

⚠ CAI EM PROVA

Complexidade ciclomática de McCabe indica o número de caminhos independentes.

Material de estudo e revisão · uso pessoal

TDD e BDD

O QUE É

Abordagens em que testes ou comportamentos guiam o desenvolvimento.

1 TDD ciclo

- vermelho
- verde
- refatorar

2 TDD regra

- escrever o teste antes do código

3 BDD

- comportamento descrito em linguagem natural

4 Gherkin

- Dado
- Quando
- Então

5 Benefícios

- design melhor
- menos defeitos
- documentação viva

⚠ CAI EM PROVA

No TDD o teste falha primeiro (vermelho) antes da implementação.

Material de estudo e revisão · uso pessoal

Qualidade: ISO/IEC 25010

O QUE É

Modelo de qualidade de produto de software da família SQuaRE.

1 Características

- adequação funcional
- eficiência
- compatibilidade
- usabilidade

2 Mais

- confiabilidade
- segurança
- manutenibilidade
- portabilidade

3 Qualidade em uso

- efetividade
- eficiência
- satisfação
- ausência de risco

4 Família SQuaRE

- substitui a ISO 9126

5 Manutenibilidade

- modularidade
- reusabilidade
- testabilidade

⚠ CAI EM PROVA

A 25010 substituiu a ISO 9126 e incluiu segurança e compatibilidade.

Material de estudo e revisão · uso pessoal

CMMI

O QUE É

Modelo de maturidade de processos para desenvolvimento e serviços.

1 Níveis de maturidade

- inicial
- gerenciado
- definido
- quantitativamente gerenciado
- em otimização

2 Representações

- por estágios (maturidade)
- contínua (capacidade)

3 Nível 2

- gestão básica de projetos

4 Nível 3

- processos padronizados na organização

5 Nível 5

- melhoria contínua com base quantitativa

⚠ CAI EM PROVA

O nível 1 não exige nenhuma área de processo.

Material de estudo e revisão · uso pessoal

MPS.BR

O QUE É

Modelo brasileiro de melhoria de processo de software, da Softex.

1 Níveis

- de G (menor) a A (maior)

2 Total

- 7 níveis de maturidade

3 Nível G

- gerência de projetos e de requisitos

4 Compatível

- com CMMI e normas ISO

5 Foco

- micro, pequenas e médias empresas

⚠ CAI EM PROVA

No MPS.BR a ordem é invertida: A é o nível mais alto.

Análise de Pontos de Função

O QUE É

Mede o tamanho funcional do software do ponto de vista do usuário.

1 Funções de dados

- ALI (interno)
- AIE (externo)

2 Funções de transação

- EE
- SE
- CE

3 Contagem

- não ajustada
- fator de ajuste (opcional)

4 Tipos de contagem

- desenvolvimento
- melhoria
- aplicação

5 Órgão

- IFPUG e manual CPM

⚠ CAI EM PROVA

APF independe da linguagem e da tecnologia usadas.

Material de estudo e revisão · uso pessoal

Estimativas e Métricas

O QUE É

Técnicas para prever esforço, prazo e custo do software.

1 COCOMO II

- estima esforço por tamanho e fatores de custo

2 Planning Poker

- estimativa ágil em pontos de história

3 Analogia

- compara com projetos anteriores

4 Métricas de processo

- produtividade
- defeitos por KLOC

5 Velocidade

- pontos entregues por sprint

⚠ CAI EM PROVA

Pontos de história medem esforço relativo, não horas.

Gerência de Configuração

O QUE É

Controla versões e mudanças de itens de configuração do software.

1 Atividades

- identificação
- controle de mudança
- auditoria
- relato de status

2 Item de configuração

- código
- documento
- ambiente

3 Baseline

- versão aprovada de referência

4 Ferramentas

- Git
- SVN

5 Branch e merge

- desenvolvimento paralelo

⚠ CAI EM PROVA

Baseline só muda por processo formal de controle de mudança.

Material de estudo e revisão · uso pessoal

Manutenção de Software

O QUE É

Modificações feitas no software depois da entrega.

1 Corretiva

- corrige defeitos

2 Adaptativa

- ajusta a novo ambiente

3 Perfectiva

- melhora ou acrescenta funcionalidades

4 Preventiva

- evita problemas futuros

5 Leis de Lehman

- o software precisa evoluir ou perde utilidade

⚠ CAI EM PROVA

Adaptar o sistema a um novo sistema operacional é manutenção adaptativa.

Arquitetura de Software

O QUE É

Estrutura de alto nível do sistema: componentes, relações e princípios.

1 Camadas

- apresentação
- negócio
- dados

2 Cliente-servidor

- requisições a um servidor central

3 Microsserviços

- serviços pequenos e independentes

4 Orientada a eventos

- produtores e consumidores

5 Atributos

- desempenho
- escalabilidade
- manutenibilidade

⚠ CAI EM PROVA

Em camadas estritas, cada camada só fala com a imediatamente abaixo.

Material de estudo e revisão · uso pessoal

Engenharia Reversa e Reengenharia

O QUE É

Analisar um sistema existente para entender e, se preciso, reconstruir.

1 Engenharia reversa

- do código para modelos e projeto

2 Reengenharia

- reestrutura e reconstrói o sistema

3 Refatoração

- muda a estrutura sem mudar o comportamento

4 Motivo

- sistema legado
- documentação perdida

5 Engenharia avante

- do projeto para o código

⚠ CAI EM PROVA

Refatorar não altera o comportamento externo do software.

Material de estudo e revisão · uso pessoal