

Resumo Engenharia de Software I — Prof. Agnaldo Cieslak

Resumo para prova do Ciclo 1

Competências da disciplina

- Analisar e especificar projetos de software.
 - Desenhar projetos seguindo fases do ciclo de vida.
 - Elicitar (coletar) requisitos do software.
 - Especificar requisitos funcionais, não funcionais e inversos.
 - Construir modelo e diagramas de casos de uso.
 - Elaborar User Stories.
 - Documentar diagramas de casos de uso e de atividades.
-

Ementa (Temas principais)

- Ciclo de vida de desenvolvimento de software.
 - Processo de desenvolvimento.
 - Elicitação de requisitos.
 - Requisitos Funcionais, Não-Funcionais e Inversos.
 - Análise de requisitos com metodologias ágeis.
 - User Stories e Story points.
 - UML: Diagrama e especificação de Casos de Uso.
 - UML: Diagrama de Atividades.
-

Ciclo de Vida de Desenvolvimento de Software (SDLC)

- **Definição:** Conjunto de fases e atividades organizadas para planejar, controlar e gerenciar projetos de software.
- Cada fase gera produtos que alimentam a fase seguinte.
- Permite planejamento com marcos de controle.

Técnicas

- **Fast-tracking:** iniciar uma atividade antes da anterior acabar, com cuidado nos riscos.
 - **Collision:** alocar mais pessoas para acelerar uma tarefa, tratando riscos.
-

Fases do SDLC

1. Planejamento:

- Análise de custo-benefício, programação, alocação de recursos.
- Coleta requisitos de clientes e partes interessadas.
- Documento de especificação de requisitos.

2. Projeto (Design):

- Análise dos requisitos e definição das melhores soluções.
- Escolha de tecnologias e integração com sistemas existentes.

3. Implementação:

- Codificação do software, tarefa por tarefa.

4. Teste:

- Identificação de bugs com testes manuais e automatizados.
- Verificação do atendimento aos requisitos.

5. Implantação:

- Preparação do ambiente de produção.
- Configuração e instalação do software para o usuário final.

6. Manutenção:

- Correção de erros, atualizações e melhorias baseadas no uso real.
-

Modelos de Ciclo de Vida

- **Sequencial (Cascata):**

- Fases em ordem fixa, uma após a outra.
- Foco em disciplina e revisão formal.

- **Iterativo (Incremental):**

- Desenvolvimento em pequenos passos (iterações).
- Fases com múltiplas iterações, ajustando o produto ao longo do tempo.

- **RAD (Rapid Application Development):**

- Desenvolvimento rápido e incremental (60-90 dias).
- Modular, baseado em componentes.
- Requisitos bem definidos e escopo restrito.
- Requer equipes suficientes e forte comprometimento.

- **Evolutivo:**

- Software vai evoluindo a partir de versões parciais.
- Requisitos parcialmente conhecidos no início.
- Risco: protótipos podem ser usados como produto final.

- **Espiral:**

- Combina prototipação iterativa com cascata.

- Focado em gestão de riscos.
 - Desenvolvido em ciclos evolutivos com marcos de decisão.
 - Exige experiência em análise de riscos.
 - **RUP (Rational Unified Process):**
 - Processo configurável, iterativo-incremental.
 - Baseado em disciplinas (áreas de interesse) e papéis (responsabilidades).
 - Uso intensivo de UML e desenvolvimento dirigido por casos de uso.
 - Avaliação contínua de riscos e geração de produtos em cada iteração.
-

Conclusão da Aula 1

- Entendimento do que é o SDLC e seus diferentes modelos.
- Importância do planejamento, controle e gestão de riscos.
- Compreensão dos ciclos iterativos e a evolução do software.
- Introdução a métodos e processos usados para garantir qualidade e sucesso no desenvolvimento.