

Engenharia de Software

Ítalo Mendes da Silva Ribeiro

Universidade Federal do Ceará

23 de maio de 2019

Súmario

Introdução

Ciclo de Vida

Metodologias

Modelo em Cascata

Métodos Ágeis

UML

IHC

Súmario

Introdução

Ciclo de Vida

Metodologias

Modelo em Cascata

Métodos Ágeis

UML

IHC

Engenharia de Software é o ramo da ciência da computação que estuda princípios para guiar o desenvolvimento de grandes e complexos sistemas de software

Questões importantes no desenvolvimento de software

- ▶ Como estimar o custo em dinheiro para o desenvolvimento de um software?
- ▶ Como dividir o projeto de desenvolvimento do software em pedaços menores?
- ▶ Como garantir que os pedaços são compatíveis?

Questões importantes no desenvolvimento de software

- ▶ Como garantir a boa comunicação entre os membros da equipe de desenvolvimento?
- ▶ Como medir o progresso do desenvolvimento?
- ▶ Como lidar com os diferentes detalhes técnicos e administrativos do projeto?

Introdução

- ▶ A engenharia de software utiliza algumas ideias da engenharia civil
- ▶ Existem diferenças entre o produto desenvolvimento pela engenharia de software e a engenharia civil
- ▶ Uma das ideias da engenharia é o aproveitamento de componentes já prontos
- ▶ Embora muitos sistemas complexos são escritos sem utilizar componentes já prontos

- ▶ Existem poucas técnicas para medir quantitativamente precisamente as propriedades do software
- ▶ As propriedades de medição são chamadas de **métricas**
- ▶ As métricas medem o desenvolvimento do software mas podem dar impressões erradas do estado do desenvolvimento
- ▶ **EXEMPLO:** de métricas de desenvolvimento
 - ▶ Número de linhas de código
 - ▶ Velocidade de execução
 - ▶ Número de erros
 - ▶ Confiabilidade
 - ▶ Funcionalidade

- ▶ Análise e desenvolvimento do projeto de grandes e complexos softwares é difícil
- ▶ Ferramentas CASE (*Computer-aided Software Engineering*) auxiliam no projeto de desenvolvimento de software
- ▶ Ferramentas CASE realizam:
 - ▶ Planejamento
 - ▶ Estimativas
 - ▶ Cronogramas
 - ▶ Documentação
 - ▶ Criação de interfaces gráficas
 - ▶ Prototipagem, etc

- ▶ IDEs (*Integrated Development Environments*) são softwares que também auxiliam no desenvolvimento
- ▶ IDEs possuem:
 - ▶ Editor de texto
 - ▶ Editor de interface gráfica
 - ▶ Compilador
 - ▶ Ferramentas de debug
 - ▶ Conexão com repositório de código, etc

Súmario

Introdução

Ciclo de Vida

Metodologias

Modelo em Cascata

Métodos Ágeis

UML

IHC

Ciclo de Vida de Software

Ciclo de vida dividido em:

- ▶ Desenvolvimento (*Development*)
- ▶ Uso (*Use*)
- ▶ Manutenção (*Maintenance*)

Ciclo de Vida de Software

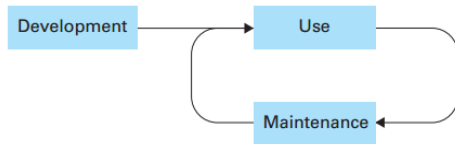


Figura: Três fases do ciclo de vida de um software.

Ciclo de Vida de Software

Ciclo de vida dividido em:

- ▶ **Em produtos normais:** a manutenção consiste no reparo
- ▶ **Em software:** manutenção correção de erros e/ou atualização
- ▶ Manutenção pode durar todo o tempo de utilização de um software

Súmario

Introdução

Ciclo de Vida

Metodologias

Modelo em Cascata

Métodos Ágeis

UML

IHC

Metodologias de Engenharia de Software

Engenharia de
Software

Ítalo Ribeiro

Introdução

Ciclo de Vida

Metodologias

–Modelo em Cascata

–Ágeis

UML

IHC

- ▶ Metodologias ou modelos determinam de que maneira o software será desenvolvido
- ▶ Principais modelos de desenvolvimento de software:
 - ▶ Modelo em cascata
 - ▶ Modelo iterativo e incremental
 - ▶ Métodos ágeis

Modelo em Cascata

- ▶ Modelo tradicional de desenvolvimento de software
- ▶ Dividido em:
 - ▶ Análise de requisitos
 - ▶ Projeto
 - ▶ Implementação
 - ▶ Teste

Modelo em Cascata

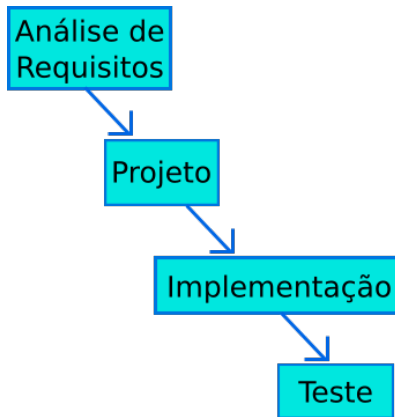


Figura: Etapas do modelo cascata de desenvolvimento de software.

Análise de requisitos

- ▶ Especificar os serviços que serão realizados pelo software que será desenvolvido
- ▶ Definir restrições dos serviços e como se comunicarão com o mundo externo
- ▶ Requisitos do software escritos no **documento de especificação de requisitos**
- ▶ O documento de requisitos é importante porque guiará o desenvolvimento do software

Análise de requisitos

- ▶ Problemas comuns da análise de requisitos:
 - ▶ Problemas de comunicação dos desenvolvedores com os clientes que contrataram a empresa para desenvolver o software
 - ▶ Mudanças dos requisitos
 - ▶ Melhorias tecnológicas podem causar mudanças nos requisitos

Projeto

- ▶ Cria o plano de desenvolvimento do software proposto
- ▶ Definição da estrutura interna do software
- ▶ Diagramas e modelos são usados para representação da estrutura e execução do software
- ▶ A notação dos diagramas e modelos muda com certa frequência

Implementação

- ▶ Escrita do código-fonte e criação do banco de dados (caso necessário)
- ▶ **Analista de software** envolvido no processo de desenvolvimento e análise de requisitos
- ▶ **Programador** responsável pela implementação em si
- ▶ Frequentemente o profissional de computação é analista e programador em um projeto

Teste

- ▶ Antigamente, testes eram realizados apenas após o desenvolvimento do software
- ▶ Atualmente, testes são realizados em cada versão incremental do software
- ▶ Ajuda na localização de problemas nas fases iniciais do desenvolvimento
- ▶ Não é possível eliminar totalmente os erros de um sistema

Problemas do modelo em cascata

- ▶ Falhas no projeto só eram detectadas na fase de implementação
- ▶ As frequentes mudanças de requisitos prejudicam o desenvolvimento do software

- ▶ Maior crescimento em utilização atualmente
- ▶ Melhor adaptação as mudanças de requisitos, pois caso os requisitos mudem, as alterações são realizadas de imediato
- ▶ Menor rigor e formalismo que outros modelos
- ▶ **EXEMPLO:** XP (*eXtreme Programming*), SCRUM, FDD (*Feature Driven Development*), etc

Programação extrema ou eXtreme Programming (XP)

- ▶ Equipes pequenas e médias
- ▶ Software com requisitos ainda incertos
- ▶ Pequenas fases de desenvolvimento
- ▶ Projeto simples
- ▶ Propriedade coletiva, ou seja, código fonte de qualquer membro da equipe
- ▶ Programação em pares

Súmario

Introdução

Ciclo de Vida

Metodologias

Modelo em Cascata

Métodos Ágeis

UML

IHC

- ▶ **UML** (*Unified Modeling Language*)
- ▶ Linguagem usada para modelagem e notação do software
- ▶ Usado principalmente em softwares desenvolvidos em linguagens orientadas a objetos
- ▶ Utiliza diagramas para representar as informações do software
- ▶ Cada diagrama representa um aspecto diferente do software

Diagrama de casos de usos

- ▶ Representa o software como atores e casos de usos
- ▶ Atores são os tipos de usuários
- ▶ Casos de usos são as ações realizadas pelos atores no software

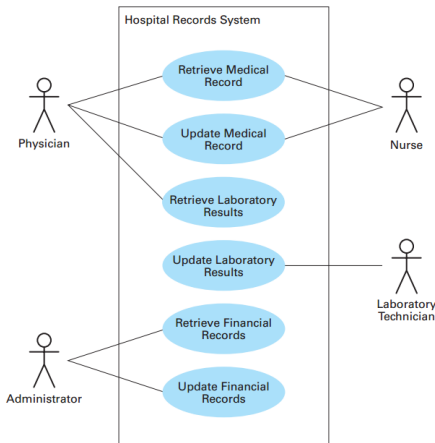


Figura: Diagrama de caso de uso da UML de um software para registro das operações de um sistema hospitalar.

Diagrama de fluxo

- ▶ Usado em softwares desenvolvidos em linguagens de paradigma imperativo
- ▶ Considera principalmente a manipulação dos dados e não os procedimentos
- ▶ O diagrama de fluxo representa o caminho das informações durante execução do programa
- ▶ Retângulos representam os dados e as setas o caminho

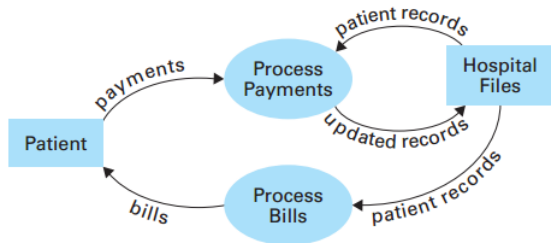


Figura: Diagrama de fluxo de dados do sistema de pagamento de pagamento de um hospital por parte do paciente.

Súmario

Introdução

Ciclo de Vida

Metodologias

Modelo em Cascata

Métodos Ágeis

UML

IHC

Interface Humano-Computador

- ▶ **IHC** (Interface Humano-Computador)
- ▶ É uma área da engenharia de software que estuda a interação dos usuários com os computadores
- ▶ Objetiva apontar as melhores formas para que o usuário possa utilizar um software ou um dispositivo
- ▶ Aborda tópicos de design, psicologia, ergonomia, sociologia, linguística, semiótica e computação

- ▶ **Usabilidade** é o termo usado para definir a facilidade de uso de um software ou dispositivo
- ▶ **Ergonomia** trata do design de software que harmonizem com as habilidades físicas dos humanos
- ▶ **Cognição** trata do design de software que considere as habilidades mentais do usuário como atenção, percepção, memória e raciocínio
- ▶ **EXEMPLO:** disposição das teclas na máquina de escrever ou no teclado. As letras mais usadas estão nas teclas guias (mais próximas da posição de descanso dos dedos)

Técnicas de Modelagem e Sistemas de Notação

Engenharia de
Software

Ítalo Ribeiro

Introdução

Ciclo de Vida

Metodologias

–Modelo em Cascata

–Ágeis

UML

IHC



Figura: Imagem de um teclado, onde as teclas guias estão destacadas pelos retângulos vermelhos.

- ▶ A interface gráfica de um software deve ser construída utilizando várias recomendações da IHC
- ▶ Software com interface gráfica ruim leva ao descontentamento por parte do usuário
- ▶ Exemplos de recomendações para interfaces gráficas:
 - ▶ Evitar a poluição visual da tela
 - ▶ Principais informações visualizadas facilmente
 - ▶ Organização do conteúdo de maneira organizada
 - ▶ Máximo de 3 ou 4 cores
 - ▶ Botões grandes para telas de aplicativos para smartphones

Técnicas de Modelagem e Sistemas de Notação

Engenharia de
Software

Ítalo Ribeiro

Introdução

Ciclo de Vida

Metodologias

–Modelo em Cascata

–Ágeis

UML

IHC

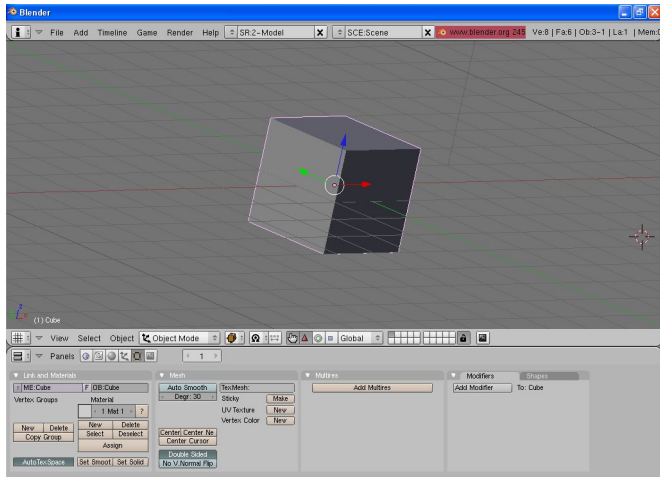


Figura: Antiga interface do Blender com muitos botões de tamanhos diferentes, textos e ícones diferentes.

Técnicas de Modelagem e Sistemas de Notação

Engenharia de
Software

Ítalo Ribeiro

Introdução

Ciclo de Vida

Metodologias

–Modelo em Cascata

–Ágeis

UML

IHC

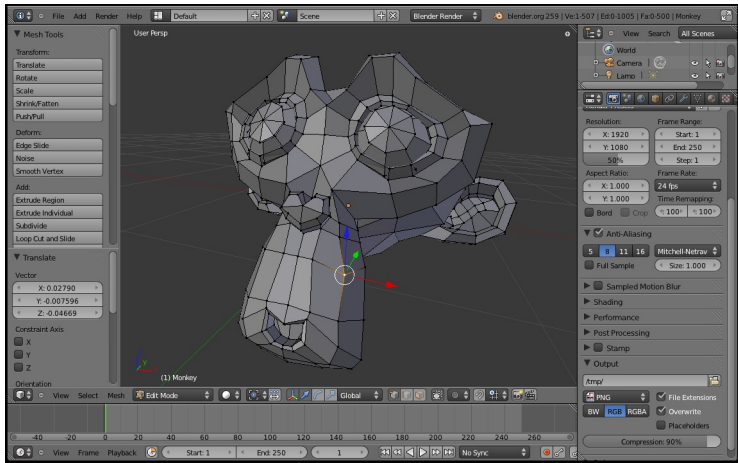


Figura: Interface atual do Blender mais organizada e mostrando apenas o conteúdo necessário no momento.