Projeto Integrado III

DD - UFC - Quixadá

Engenharia de Software: Conceitos Básicos - Parte 1

Prof.: Aníbal Cavalcante

Problema — Edsger Dijkstra, The Humble Programmer

1. Crise do Software nos anos 70.
2. Por que os sistemas informatizados:
    a. não faziam o que deveriam fazer?
    b. eram entregues com atraso?
    c. custavam mais caro do que o previsto?
    d. eram de baixa qualidade?
    e. eram pouco confiáveis?
    f. eram(são) lentos?
    g. difíceis de usar?

Programa de Computador vs Software

1. Software é mais do que um único programa.
2. Um sistema profissional, pode consistir em:
    a. Uma série de programas separados que trabalham em conjunto e seus diversos arquivos de configuração.
    b. Documentações, que descrevem a estrutura do sistema como um todo;
    c. Documentações, voltadas ao usuário final desse sistema, explicando suas instruções de uso;
    d. Páginas Web para que os usuários façam downloads de correções e atualizações do produto.

Programa de Computador vs Software

Diferença:

a. Em um programa escrito para uso pessoal, ninguém mais vai usá-lo, assim o desenvolvedor não precisa se preocupar em escrever guias de programação, módulos distribuídos, separação de responsabilidades, reusabilidade, manutenibilidade e etc.
b. Já um software, outras pessoas irão utilizar, outros engenheiros terão que mantê-lo, mudá-lo ou fazer pequenos ajustes, assim, geralmente o desenvolvedor terá que fornecer informações adicionais, além do próprio código-fonte desse sistema.

Nascimento da Engenharia de Software

- Vários processos e metodologias nasceram ao longo das últimas décadas para melhorar e produzir software de qualidade e cumprindo prazos e orçamentos.
- Por exemplo:
  - Programação Orientada a Objetos e novas linguagens...
  - Desenvolvimento Orientado a Testes….
  - Desenvolvimento Incremental e Iterativo...
  - Modelos de Processo de Software….
  - Padrões Internacionais de Processo de Software... (ISO 12207)
  - Padrões Internacionais de Qualidade de Software…. (ISO 9126)

O que é Engenharia de Software?

É uma área da computação voltada para:

- especificação;
- desenvolvimento;
- e manutenção de software.

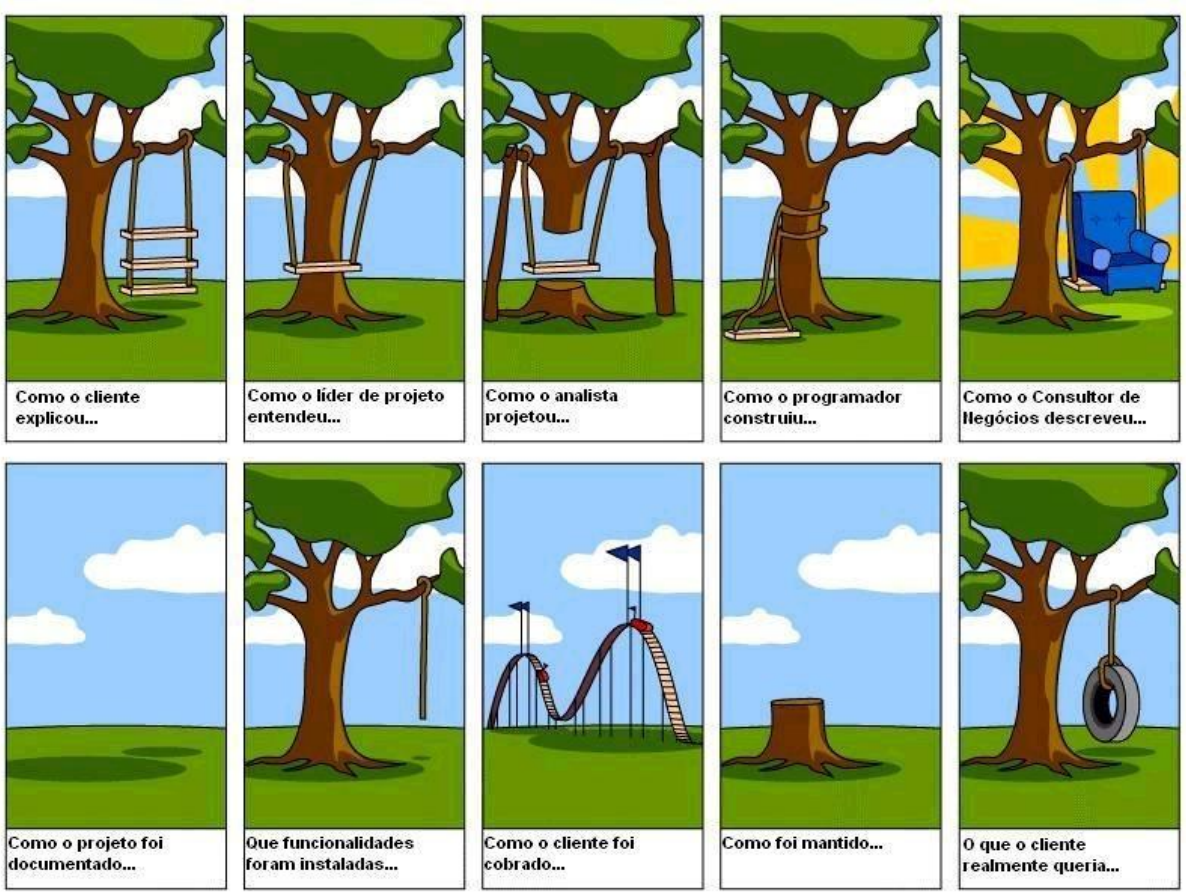
Com aplicação de tecnologias e práticas de gerência de projetos e outras disciplinas, visando:

1. organização;
2. produtividade;
3. qualidade.

Áreas de conhecimento (IEEE - SWEBOK - (documento))

1. Requisitos de software
2. Projeto de software
3. Construção de software
4. Teste de software
5. Manutenção de software
6. Gerência de configuração de software
7. Gerência de engenharia de software
8. Processos de Engenharia de Software
9. Ferramentas e Métodos de Engenharia de Software
10. Qualidade de software

O que é um Software de Qualidade?



Características de um Software de Qualidade

Característica	Descrição
Manutenibilidade	Um software deve ser escrito de tal forma que ele possa evoluir para atender às necessidades em constante mudança dos clientes. Este é um atributo fundamental, porque a mudança de software é uma exigência inevitável de um ambiente de negócios em constante mudança.
Confiabilidade e segurança	Confiabilidade de software inclui uma série de características, incluindo a confiabilidade e segurança. Software confiável não deve causar dano físico ou econômico, no caso de falha do sistema. Usuários mal-intencionados não devem ser capazes de acessar ou danificar o sistema.

Características de um Software de Qualidade

Característica	Descrição
Eficiência	Um software eficiente não deve desperdiçar os recursos do sistema, como memória e ciclos do processador. Eficiência inclui, portanto, a capacidade de resposta, tempo de processamento, utilização de memória, etc...
Aceitabilidade	Um software deve ser aceitável para os tipos de usuários para o qual foi concebido. Isso significa que ele deve ser compreensível, usável e compatível com outros sistemas que estes usuários utilizam.

Processos de Software

Definição 1: Uma abordagem sistemática usada na engenharia de software para criação de produtos de software.

Definição 2: Uma sequência de atividades que leva à produção de um produto de software.

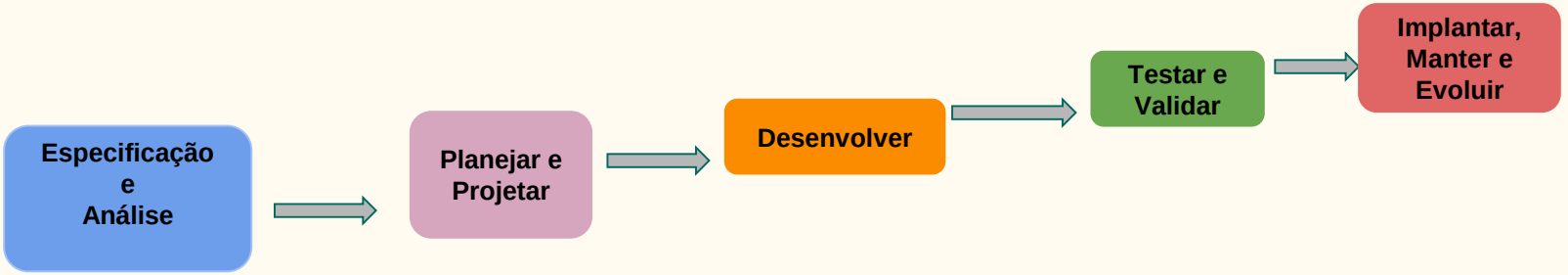
Definição 3: A divisão do trabalho de desenvolvimento de software em fases distintas (ou estágios) contendo atividades com a intenção de melhorar planejamento e gerenciamento.

Processos de Software

Existem diversos **processos de desenvolvimento de software**, além do gerenciamento existem algumas atividades básicas comuns à maioria dos processos existentes, como:

1. Especificação ou Levantamento de requisitos;
2. Análise de Requisitos;
3. Planejamento e Projeto;
4. Desenvolvimento;
5. Testes;
6. Implantação, Manutenção e Evolução.

Processos de Software (Atividades Fundamentais)



Também conhecido como:

1 - Metodologia de Desenvolvimento de Software
2 - Ciclo de vida de Software
3 - Processo de Desenvolvimento de Software

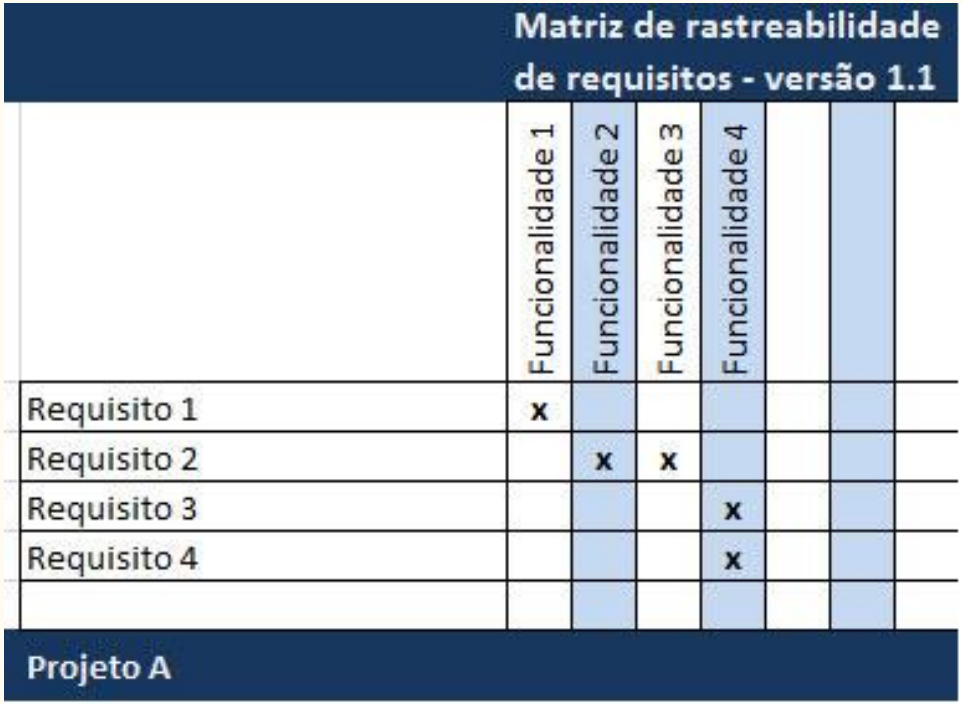
Processos de Software(Atividades Fundamentais)

1. **Levantamento ou Especificação de requisitos**
    a. Objetivo de compreender o problema;
    b. Equipe de Desenvolvimento deve possuir a mesma visão do software;
    c. Desenvolvedores e clientes, em conjunto, buscam levantar e priorizar as necessidades dos futuros usuários do software;
    d. Essas necessidades são denominadas como **requisitos**;
    e. É a etapa mais importante, no que diz respeito ao retorno de investimentos no projeto;

Processos de Software(Atividades Fundamentais)

2. **Análise de Requisitos**
    a. Os desenvolvedores fazem um estudo detalhado dos dados levantados na atividade de levantamento.
    b. Junto ao cliente, verificam se existem questões em aberto ou problemas mal compreendidos;
    c. Protótipos podem e devem ser utilizados nesta etapa.
    d. Casos de testes são criados para a montagem do documento de testes.
    e. Verificam se existem requisitos que são antagônicos, divergentes ou conflitantes;
    f. Verificam o relacionamento entre os **requisitos e as funcionalidades** do software através da **matriz de rastreabilidade de requisitos**.

Processos de Software(Atividades Fundamentais)



Matriz de rastreabilidade de requisitos - versão 1.1							
	Funcionalidade 1	Funcionalidade 2	Funcionalidade 3	Funcionalidade 4			
Requisito 1	x						
Requisito 2		x	x				
Requisito 3				x			
Requisito 4				x			
Projeto A							

Processos de Software(Atividades Fundamentais)

- **Tipos de Requisitos**
  1. **Requisitos Funcionais** referem-se sobre o que o sistema deve fazer, ou seja, suas funções e informações.
  2. Preocupam-se com a funcionalidade e os serviços do sistema, ou seja, as funções que o sistema deve fornecer para o cliente, e como o sistema se comportará em determinadas situações. Segue abaixo alguns exemplos de requisitos funcionais:

- [RF001] O Sistema deve cadastrar médicos profissionais (entrada)
- [RF002] O Sistema deve emitir um relatório de clientes (saída)
- [RF003] O Sistema deve passar um cliente da situação "em consulta" para "consultado" quando o cliente terminar de ser atendido (mudança de estado)
- [RF004] O cliente pode consultar seus dados no sistema

Processos de Software(Atividades Fundamentais)

- **Tipos de Requisitos**
  1. **Requisitos Não Funcionais** referem-se aos critérios que qualificam os requisitos funcionais.
  2. Definem propriedades e restrições do sistema como tempo, espaço, linguagens de programação, versões do compilador, BD e Sistema Operacional, etc.
  3. Devemos preferencialmente associar uma medida ou referência para cada requisito não funcional.

- [RNF001] O sistema deve imprimir o relatório em até 5 segundos.
- [RNF002] Todos os relatórios devem seguir o padrão de relatórios especificado pelo setor XYZ.
- [RNF003] O sistema deve ser implementado em Java.

Métricas de RNFs

Propriedade	Métrica
Velocidade	Transações processadas por segundo. Tempo de resposta ao usuário/evento. Tempo de atualização da tela.
Tamanho	Kbytes. Número de chips de RAM.
Facilidade de uso	Tempo de treinamento. Número de telas de ajuda.
Confiabilidade	Tempo médio para falhar. Probabilidade de indisponibilidade. Taxa de ocorrência de falhas. Disponibilidade.
Robustez	Tempo de reinício após falha. Porcentagem de eventos que causam falhas. Probabilidade de que os dados sejam corrompidos por falhas.
Portabilidade	Porcentagem de declarações dependentes de sistema-alvo. Número de sistemas-alvo.

Processos de Software(Atividades Fundamentais)

3. **Planejamento e Projeto**
    a. Definir como o sistema funcionará internamente, para que os requisitos do cliente possam ser atendidos.
    b. Definição da arquitetura do sistema, linguagem de programação utilizada, Banco de Dados, padrão de interface gráfica, entre outros.
    c. No projeto é gerada uma descrição computacional, mencionando o que o software deve fazer, e deve ser coerente com a descrição realizada na fase de análise de requisitos.

Processos de Software(Atividades Fundamentais)

4. **Desenvolvimento**
    a. Nessa etapa, o sistema é codificado a partir da descrição computacional da fase de projeto em uma outra linguagem, onde se torna possível a compilação e geração do código-executável para o desenvolvimento software.
    b. Testes de validação de componentes em ambiente de desenvolvimento podem ser feitos nessa etapa, chamados de testes alphas.

Processos de Software(Atividades Fundamentais)

5. **Testes**
    a. Diversas atividades de testes são executadas a fim de se validar o produto de software, testando cada funcionalidade de cada módulo, buscando, levando em consideração a especificação feita na fase de projeto.
    b. O principal resultado é o relatório de testes, que contém as informações relevantes sobre erros encontrados no sistema, e seu comportamento em vários aspectos.
    c. Ao final dessa atividade, os diversos módulos do sistema são integrados, resultando no produto de software.
    d. Antes de entregar e implantar a versão final para os usuários é realizado o teste beta, ou seja, em ambiente de produção, o sistema é utilizado pela primeira vez;

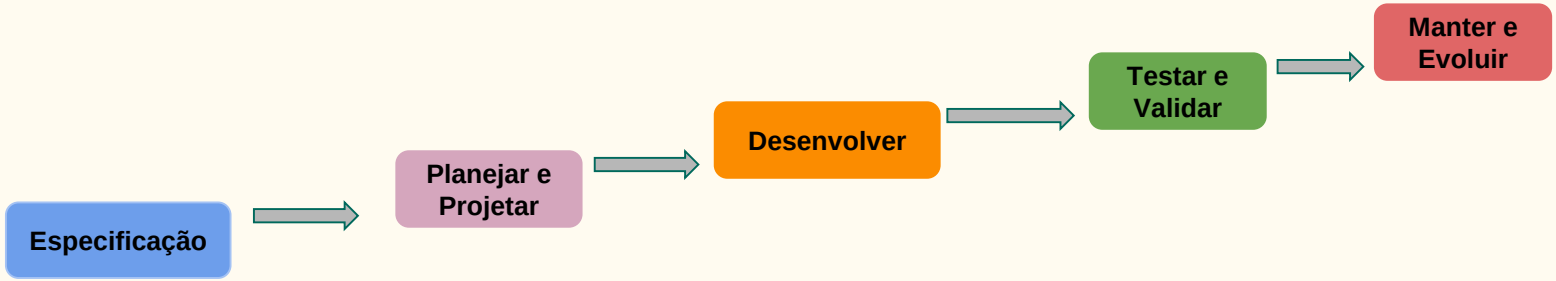
Processos de Software(Atividades Fundamentais)

6. **Implantar, Manter e Evoluir**
    a. Entregar e implantar a versão final no ambiente real de produção do usuário;
    b. Corrigir eventuais bugs e realizar melhorias;
    c. Adicionar novos requisitos;

Propósitos dos Processos de Software

Quanto ao seu propósito, as atividades ou processos podem ser classificados ainda como:

1. **Atividades de Desenvolvimento:** (definir requisitos, projetar, desenvolver…..)
2. **Atividades de Gerência:** (realização de estimativas, elaboração de cronogramas, análise dos riscos do projeto etc.)
3. **Atividades de Garantia da Qualidade:** (testes unitários, revisões e inspeções de produtos)



Diversidade de Engenharia de Software

- **Engenharia de software** é uma abordagem sistemática para a produção de softwares que leva em conta **o custo prático**, **cronograma** e questões de **confiabilidade**, bem como as **necessidades dos clientes** e produtores de softwares.
- **A natureza do software indica qual a metodologia deve ser utilizada, e não o contrário disso.**
- **Por exemplo….**

Ideias Fundamentais da Engenharia de Software

1. Desenvolvidos através de um processo ou método gerenciado e compreendido.
2. Foco no planejamento tendo ideias claras sobre o que vai ser produzido e quando ele será concluído.
3. Confiabilidade e desempenho são importantes para todos os tipos de sistemas.
4. Um software deve se comportar como esperado, sem falhas e deve estar disponível para uso quando for necessário.
5. Deve ser seguro e protegido contra ataques externos.

Ideias Fundamentais da Engenharia de Software

6. O sistema deve funcionar de maneira eficiente e não deve desperdiçar recursos.
7. Compreender e gerir as especificações e requisitos de software (o que o software deve fazer).
8. Deve-se saber o que os diferentes clientes e usuários do sistema esperam dele.
9. Gerenciar essas necessidades e entregar dentro do orçamento e do prazo.
10. Fazer uso mais eficaz possível dos recursos existentes.
11. Reutilizar software que já foi desenvolvido, em vez de escrever um novo software.