

COLÉGIO ESTADUAL ULYSSES GUIMARÃES
CURSO TÉCNICO PROFISSIONALIZANTE EM INFORMÁTICA

ERINALDO SANCHES NASCIMENTO

FASES DO DESENVOLVIMENTO DE SOFTWARE

FOZ DO IGUAÇU

2013

1. DESENVOLVIMENTO DE SOFTWARE	3
1.1. MODELOS DO PROCESSO DO CICLO DE VIDA.....	5
1.1.1. <i>Concepção</i>	5
1.1.2. <i>Levantamento, exploração e modelagem de requisitos</i>	6
1.1.3. <i>Projeto</i>	7
1.1.4. <i>Codificação e depuração</i>	8
1.1.5. <i>Testes</i>	9
1.1.6. <i>Instalação</i>	9
1.1.7. <i>Manutenção e evolução do software</i>	9
1.1.8. <i>Aposentadoria</i>	10
1.2. EXERCÍCIOS	11
1.3. REFERÊNCIA BIBLIOGRÁFICA	15

1. DESENVOLVIMENTO DE SOFTWARE

Desenvolvimento de software é o processo de tomada de um conjunto de requisitos de um usuário (o enunciado do problema), analisá-los, projetar uma solução, e então implementá-la em um computador.

A programação é a parte de implementação, ou possivelmente a parte de projeto e implementação, do desenvolvimento de software. A programação é fundamental para o desenvolvimento de software, mas não é a coisa toda.

Não é engenharia de software. Engenharia de software também envolve um processo e inclui o desenvolvimento de software, mas inclui também todo o lado de gestão da criação de um programa de computador que as pessoas usarão. Engenharia de software inclui gerenciamento de projeto, gerenciamento de configuração, cronograma e avaliação, construção da linha de base e cronograma, gerenciamento de pessoas, e várias outras coisas.

O desenvolvimento de software é um estreitamento do foco da engenharia de software no que diz respeito à criação do software real. E é uma ampliação do foco da programação que inclui questões de análise, projeto e implantação funcional.

Depois de 60 anos ou mais de uso de computadores, descobriu-se que:

- desenvolvimento software é difícil;
- aprender a desenvolver software de forma correta, eficiente e bonita é difícil;
- é uma habilidade que você precisa adquirir e praticar (bastante);
- não se aprende programação e desenvolvimento em livros;
- aprende-se fazendo; trabalhando em problemas interessantes e difíceis.

Há várias maneiras de aprender o desenvolvimento de software. Mas todas envolvem:

- ler excelentes projetos,
- ler muitos códigos,
- escrever muitos códigos,
- pensar profundamente sobre como se aproximar de um problema e projetar a solução para ele.

O desenvolvimento de software é apenas uma parte da engenharia de software, e é o coração de cada projeto de software. A fim de fazer bem o desenvolvimento de software é necessário o seguinte:

- Uma equipe pequena e bem integrada. Manter uma equipe em vários projetos é um trabalho importante do gerente da equipe. A chave para manter uma equipe é dar-lhes um trabalho interessante a fazer e depois deixá-los sozinhos.
- A boa comunicação entre os membros da equipe. Equipes espalhados geograficamente é um grande problema nas grandes empresas de desenvolvimento de software.
- Boa comunicação entre a equipe e o cliente. Comunicação com o cliente é essencial para controlar os requisitos e exigências rotatividade durante um projeto. Clientes no local ou perto permitem a interação constante com a equipe de desenvolvimento. Os clientes podem dar um feedback imediato sobre novos lançamentos e estarem envolvidos na criação do sistema e testes de aceitação para o produto.
- Um processo que todos compreendam. Projetos e equipes maiores exigem mais coordenação e controles mais rigorosos sobre comunicação e gestão de configuração. Projetos e equipes menores tendem a seguir os processos de desenvolvimento mais ágeis, com mais flexibilidade e menos documentação exigida.
- A capacidade de ser flexível. Requisitos mudam, as pessoas vêm e vão, as ferramentas não funcionam, e assim por diante. Se você identificar riscos, tenha um plano de contingência para lidar com o evento onde o risco de fato ocorre.
- Um plano. O plano do projeto engloba o que você vai fazer para implementar o seu projeto: processo, riscos, recursos, ferramentas, gerenciamento de requisitos, estimativas, cronogramas, gerenciamento de configuração e entrega.
- Saber onde você está em todos os momentos. Equipes pequenas terão reuniões diárias para sincronizar-se no início de cada dia. Processos ágeis muitas vezes exigem reuniões diárias para melhorar a comunicação entre membros da equipe e criar um sentido de camaradagem dentro da equipe.

- Ser corajoso o suficiente para dizer: "Ei, nós estamos atrasados!" Os desenvolvedores de software são um bando de otimistas, em geral, e em algum momento você vai estar atrasado. A melhor coisa a fazer quanto a isso é dizer ao seu gerente imediato. Quanto mais cedo você descobrir que está atrasado planejar: alongamento do cronograma (improvável), mudar alguns requisitos para um lançamento futuro, obter ajuda adicional, etc. O importante é manter o gerente informado.
- As ferramentas e as práticas corretas para este projeto. Para cada projeto é preciso analisar e escolher o melhor conjunto de ferramentas de desenvolvimento para este projeto em particular. Os três fatores mais importantes na escolha de ferramentas são o tipo de aplicação que você está escrevendo, a plataforma de destino, e a plataforma de desenvolvimento. Um quarto fator na escolha da ferramenta é a composição e experiência da equipe de desenvolvimento.
- Perceber que você não sabe tudo que precisa saber no início do projeto. Você sempre vai descobrir novas exigências. Gerenciamento mudanças de requisitos durante um projeto é uma das habilidades mais importante que um desenvolvedor de software precisa ter.

1.1. MODELOS DO PROCESSO DO CICLO DE VIDA

Cada programa tem um ciclo de vida. Não importa quão grande ou pequeno o programa é, ou quantas pessoas estão trabalhando no projeto - todos os programas de percorrer os mesmos passos:

1.1.1. Concepção

A fase de planejamento é o processo fundamental de compreensão por que um sistema de informação deve ser construído e determinar como a equipe do projeto o construirá. Tem dois passos:

- I) Durante o início do projeto, o valor de negócios do sistema para a organização é identificado: como é que vai reduzir os custos ou aumentar receitas? A solicitação do sistema apresenta um breve resumo de uma necessidade de negócio, e explica como um sistema que suporta a necessidade criará valor aos negócios. O departamento de Sistema de Informação trabalha em conjunto com a pessoa ou departamento que gerou o pedido (stackholder) para realizar uma análise de viabilidade. A viabilidade analisa os aspectos fundamentais do projeto proposto:

- Viabilidade técnica da idéia (Podemos construí-lo?)
- Viabilidade econômica (Será que vai fornecer o valor do negócio?)
- Viabilidade organizacional (Se nós o construirmos, ele vai ser usado?)

O pedido do sistema e a análise de viabilidade são apresentados a uma comissão de aprovação, que decide se o projeto deve ser empreendido.

- II) Uma vez que o projeto seja aprovado, entra em gerenciamento de projetos. Durante a gestão de projetos, o gerente de projeto cria um plano de trabalho, equipes do projeto. O produto para gerenciamento de projetos é um plano de projeto, que descreve como a equipe do projeto vai começar a desenvolver o sistema.

1.1.2. Levantamento, exploração e modelagem de requisitos

A fase de análise responde as perguntas de quem vai usar o sistema, o que o sistema irá fazer, onde e quando será utilizado. Durante esta fase, a equipe do projeto investiga algum sistema atual (is), identifica oportunidades de melhoria, e desenvolve um conceito para o novo sistema.

Essa fase tem três etapas:

- I) Uma estratégia de análise é desenvolvida para orientar os esforços da equipe do projeto. Tal estratégia geralmente inclui uma análise do sistema atual e seus problemas, e, em seguida, maneiras de projetar um novo sistema.
- II) O próximo passo é levantamento de requisitos (através de entrevistas ou questionários). A análise destas informações leva ao desenvolvimento de um conceito para um novo sistema. O conceito de sistema é então usado como uma base para desenvolver um conjunto de modelos de análise de negócios, que descrevem como a empresa irá operar se o novo sistema for desenvolvido. O conjunto de modelos inclui a representação dos dados e processos necessários para apoiar o processo de negócio subjacente.
- III) As análises, o conceito de sistema e modelos são combinados em um documento chamado a proposta do sistema, que é apresentado ao patrocinador do projeto e outros tomadores de decisão (stackholders), que determinará se o projecto deve continuar a avançar.

O resultado final da fase de análise é uma análise e um projeto inicial de alto nível para o novo sistema.

1.1.3. Projeto

A fase de projeto decide *como* o sistema irá operar, em termos de:

- hardware, software e infra-estrutura de rede;
- a interface de usuário, formulários e relatórios;
- os programas específicos, bancos de dados e arquivos que serão necessários.

A fase de projeto tem quatro etapas:

- I) Esclarece se o sistema será desenvolvido por programadores da própria empresa, se o sistema será terceirizado (uma empresa

de consultoria), ou se a empresa vai comprar um pacote de software existente.

- II) Desenvolvimento do projeto da arquitetura básica para o sistema, que descreve o hardware, software, e infra-estrutura de rede para ser usado.
- III) As especificações do banco de dados e arquivo são desenvolvidas. Definem exatamente que dados serão armazenados e onde.
- IV) A equipe de analistas desenvolve o projeto do programa, definem os programas que precisam ser escritos e o que cada um faz exatamente.

Esta coleção de produtos é a especificação do sistema que é entregue à equipe de programação para a implementação. No final da fase de projeto, análise de viabilidade e plano de projeto são reexaminados e revistos, e outra decisão é tomada pelos *stackholders* sobre a possibilidade de encerrar o projeto ou continuar.

1.1.4. Codificação e depuração

A produção deve garantir a execução bem sucedida de todos os aspectos de desempenho de sistemas. Durante a produção, há a necessidade de se estabelecer como os problemas serão atendidos, que o pessoal de apoio estará disponível, e quando e como as perguntas serão respondidas e resolvidas. Este componente de produção pode iniciar novos ciclos de desenvolvimento e testes por causa de necessidades de redesenho. Isto significa que os requisitos originais não traduzem corretamente as realidades do sistema.

1.1.5. Testes

O sistema é construído e testado para garantir que funcione como projetado. Por conta do custo de bugs poder ser imenso, o teste é um dos passos mais críticos na implementação. A maioria das organizações leva mais tempo e atenção ao teste do que para escrever os programas em primeiro lugar.

1.1.6. Instalação

A instalação é o processo pelo qual o antigo sistema é desligado e o novo é ativado. Pode incluir três abordagens de transição:

- Direta: o novo sistema substitui imediatamente o sistema antigo
- Conversão paralelo: ambos os sistemas, o antigos e o novo são operados por um ou dois meses até que seja evidente que não existam erros no sistema novo
- Conversão escalonada: o novo sistema é instalado em uma parte da organização como um ensaio inicial e depois, gradualmente, instalado em outros.

Um dos aspectos mais importantes de conversão é o desenvolvimento de um plano de treinamento para ensinar os usuários utilizarem o novo sistema e ajudar a gerenciar as mudanças provocadas pelo novo sistema.

1.1.7. Manutenção e evolução do software

A equipe de analistas estabelece um plano de suporte ao sistema. Este plano geralmente inclui uma revisão formal ou informal de pós-implementação, bem como uma maneira sistemática para identificar as alterações maiores e menores necessários para o sistema.

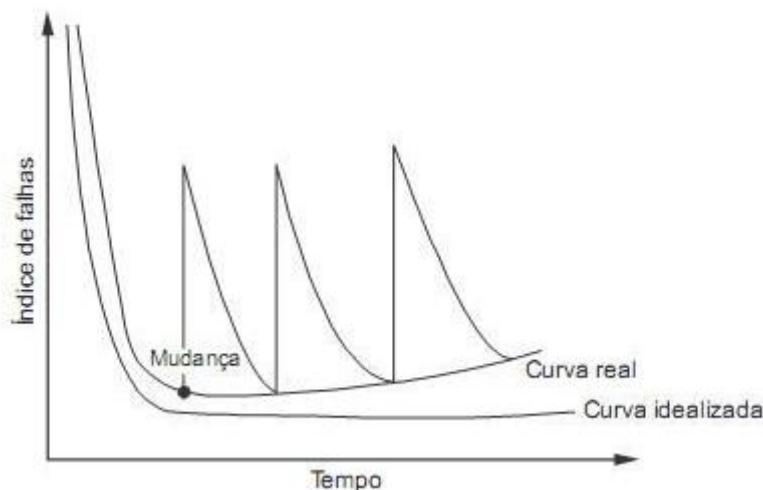
1.1.8. Aposentadoria

Um programa pode comprimir alguns desses passos, ou combinar duas ou mais etapas em uma única fase do trabalho, mas todos os programas passam por todas as etapas.

Embora cada programa tenha um ciclo de vida, existem muitas variações nos processos que envolvem estas etapas. Cada modelo de ciclo de vida é uma variação em dois tipos fundamentais. No primeiro tipo, a equipe do projeto fará um ciclo de vida completo – pelo menos as etapas 2 a 7 – antes de voltar e começar a próxima versão do produto. No segundo tipo, que prevalece nos dias de hoje, a equipe do projeto fará um ciclo de vida parcial – geralmente as etapas de 3 a 5 – e percorre esses passos várias vezes antes de prosseguir para a etapa de instalação.

1.2. EXERCÍCIOS

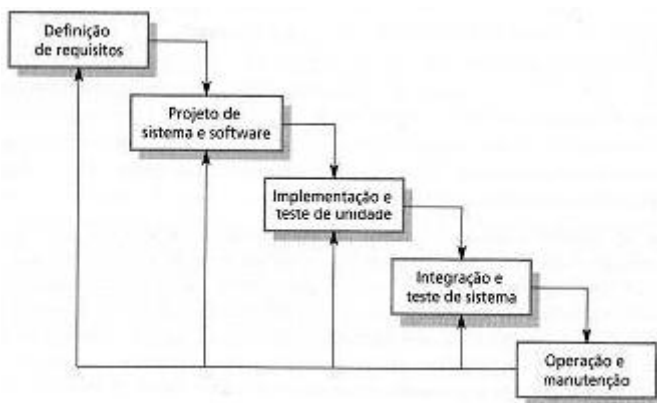
- (TCM-GO – 2012) De acordo com a engenharia de software, como todo produto industrial, o software possui um ciclo de vida. Cada fase do ciclo de vida possui divisões e subdivisões. Em qual fase avaliamos a necessidade de evolução dos softwares em funcionamento para novas plataformas operacionais ou para a incorporação de novos requisitos?
 - Fase de operação;
 - Fase de retirada;
 - Fase de definição;
 - Fase de design;
 - Fase de desenvolvimento;
- (TRT-PE – 2012) Considere o gráfico a seguir que mostra as curvas de taxa de falhas, idealizada e real.



No gráfico, a curva

- idealizada se refere à taxa de falhas do *software* e considera o processo de manutenção e mudanças do *software* no seu ciclo de vida.
- idealizada e a curva real se referem à taxa de falhas do *software*. A curva real mostra que as sucessivas alterações do *software* tendem a introduzir mais erros antes da estabilização dos erros de alterações anteriores, ocasionando a tendência crescente do índice de falhas.
- idealizada se refere ao *hardware*, que tem um alto índice de falhas no início do seu ciclo de vida ocasionadas por defeitos de fabricação e projeto, mas depois se estabiliza até o final da vida do produto.
- real se refere ao *hardware* que, desde o início do ciclo de vida do produto, pode apresentar problemas relacionados ao envelhecimento, acúmulo de poeira, vibração, abuso, temperaturas extremas, entre outros.
- idealizada e a curva real se referem à taxa de falhas do *hardware*, produto lógico, que não se desgasta, mas se deteriora rapidamente em função da introdução de erros oriundos de atividades de manutenção.

3. (TRT-PE – 2012) Considere: *é uma disciplina que se ocupa de todos os aspectos da produção de software, desde os estágios iniciais de especificação do sistema até a manutenção desse sistema, depois que ele entrou em operação. Seu principal objetivo é fornecer uma estrutura metodológica para a construção de software com alta qualidade.* A definição refere-se
- ao ciclo de vida do *software*.
 - à programação orientada a objetos.
 - à análise de sistemas.
 - à engenharia de requisitos.
 - à engenharia de *software*.
4. (TRE-SP – 2012) Durante a fase inicial do ciclo de vida do desenvolvimento de sistemas, na etapa de investigação, a tarefa que determina a probabilidade de sucesso do sistema proposto e propicia uma avaliação superficial da área técnica, econômica e comportamental do projeto, sendo decisivamente importante para o processo do desenvolvimento de sistemas é chamada
- Estudo de caso.
 - Análise de requisitos.
 - Análise de equilíbrio.
 - Estudo de viabilidade.
 - Design* lógico do sistema.
5. (TSE – 2012) Observe a figura, que representa uma ferramenta de processo, conhecida como Ciclo de Vida de Sistema. Devido ao encadeamento de uma fase com outra, esse modelo é conhecido como “cascata”. Observe.



Um das fases prevê a execução de atividades que envolvem a identificação e a descrição das abstrações fundamentais do sistema de software e suas relações e o estabelecimento de uma arquitetura geral para o sistema como um todo. Essa fase denomina-se

- definição de requisitos.
- projeto de sistema e software.
- implementação e teste de unidade.
- integração e teste de sistema.

6. (TJ-PE – 2012) O objetivo do Processo de Aquisição, um dos processos relacionados ao ciclo de vida de um *software*, é o de obter um produto ou serviço que satisfaça a necessidade do adquirente. Um dos resultados esperados deste processo é a
- a) definição das regras de auditoria do serviço.
 - b) definição dos critérios de aceitação do serviço.
 - c) lista de requerimentos de infraestrutura.
 - d) definição e a alocação de recursos e orçamentos para cada projeto.
 - e) definição das habilidades que serão necessárias para o projeto.
7. (CFA – 2010) O paradigma do ciclo de vida clássico da engenharia de *software* abrange 6 atividades. Na atividade de _____ são traduzidas as exigências de uma representação do *software* que podem ser avaliadas quanto à qualidade antes que se inicie a codificação. Assinale a alternativa que preenche corretamente a lacuna da sentença anterior.
- a) Projeto
 - b) Engenharia de sistemas
 - c) Análise e coleta de requisitos
 - d) Teste e análise de riscos
8. (UFPR – 2010) Considerando o ciclo de vida de sistemas, são processos fundamentais, de acordo com a norma ISO/IEC 12207:
- a) documentação, garantia da qualidade e verificação.
 - b) auditoria, resolução de problemas e validação.
 - c) gerência, infraestrutura, melhoria e treinamento.
 - d) gerência de configuração, fornecimento e revisão conjunta.
 - e) aquisição, desenvolvimento, operação e manutenção.
9. (Petrobrás – 2010) Uma das principais atividades do processo de teste de um ciclo de vida de um projeto qualquer é
- a) projetar testes que tratem da especificação de procedimentos externos ao computador, tais como: captação das informações, identificação das partes interessadas e distribuição das saídas.
 - b) projetar o processo de teste criando casos de teste, rotinas de teste e, eventualmente, desenvolvendo programas que fazem o teste de forma automática.
 - c) analisar e definir testes através da manipulação de ferramentas de processos usadas especialmente para obtenção de requisitos de teste de software, tais como: CMMI, BPM e ISO 9001:2000.
 - d) produzir testes e o manual de especificação do uso do sistema que é utilizado para ensinar o usuário a manipular o produto final do software.
 - e) testar as unidade de software na fase de operação e manutenção do sistema e utilizar os resultados como métricas para eventuais ajustes em projetos anteriores.
10. (Receita Federal – 2005) Analise as seguintes afirmações relacionadas a Ciclo de Vida de um projeto de software:

- I. As fases do ciclo de vida de um projeto são equivalentes aos processos de gerenciamento de projetos.
- II. As organizações consideram o ciclo de vida do produto parte do ciclo de vida do projeto.
- III. O ciclo de vida do projeto define as fases que conectam o início de um projeto ao seu final.
- IV. A transição de uma fase para a outra dentro do ciclo de vida de um projeto, em geral é definida por alguma forma de transferência técnica ou entrega.

Indique a opção que contenha todas as afirmações verdadeiras.

- a) III e IV
- b) II e III
- c) I e II
- d) I e III
- e) II e IV

1.3. REFERÊNCIA BIBLIOGRÁFICA

DOOLEY, John. **Software Development and Professional Practice**. 1st ed. New York: Apress, 2011 p. 1-7.

DENNIS, Alan; WIXOM, **Barbara H.**; **TEGARDEN, David**. Systems Analysis Design: An Object-Oriented Approach. 3rd ed. Hoboken, 2009 p. 4-6.

LANGER, Arthur M. **Guide to Software Development: Designing and Managing the Life Cycle**. 1st ed. New York: Springer, 2012 p. 1-10.