

A IMPORTÂNCIA DA ENGENHARIA DE SOFTWARE NA SOFTWARE HOUSE

Letícia Rosa de Oliveira¹

Hamanda Mendonça Ribeiro²

RESUMO

O assunto abordado nesse artigo é sobre a importância da engenharia de software na software house, ela é importante porque é através dela que o software é desenvolvido testado, qualificado, possibilitando ao profissional uma base para construção de software de alta qualidade produtiva. Sintetizando a história, conceito, a qualidade do software e processo de software. A engenharia de software identifica através de um desenvolvedor de software interfaces que devem ser estabelecidas, restrições de projetos que existem e quais critérios de validação são exigidos. Qualidade de software é um processo sistemático que focaliza todas as etapas e partes produzidas com o objetivo de garantir a conformidade de processos e produtos, prevenindo e eliminando os defeitos. Destacando como a engenharia contribui para se obter softwares de qualidades e que atenda as expectativas e necessidades de seus clientes dando ênfase na qualidade conceituando software, engenharia de software a importância da Engenharia, e da qualidade na criação de software o processo e seus ciclos usaremos o estudo bibliográfico metodologia de pesquisa bibliográfica. A engenharia de software nas software house é de suma importância pois deve-se observar as necessidades e perspectivas dos clientes começando sempre pelo planejamento, levantamento de requisitos, prototipação ,desenvolvimento, testes e manutenção do software

Palavras chave: Engenharia de software. Qualidade de software. Processo de software.

¹ Graduanda do curso de Sistemas de Informação da Faculdade de Ciências Sociais Aplicadas do Vale do São Lourenço- Eduvale

² Prof.(a)Especialista do curso de Sistemas de Informação da Faculdade de Ciências Sociais Aplicadas do Vale do São Lourenço- Eduvale

ABSTRACT

The issue addressed in this article is about the importance of software engineering in house software, it is important because it is through it that the software is developed, tested, qualified, enabling the provider a basis for high production quality software construction. We summarize the history, concept, software quality and software process. Software engineering identifies through a software developer interfaces that must be established, project constraints that exist and what validation criteria are required. Software engineering methods demonstrate the details of how to build software. They involve a wide range of tasks including: planning, project estimation, software requirement analysis and systems, maintenance and test structure design. Software quality is a systematic process that focuses on all stages and parts produced in order to ensure compliance of processes and products, preventing and eliminating defects. Highlighting how engineering contributes to obtain quality software and that meets the expectations and needs of its customers will give emphasis on quality, conceptualizing software, software engineering the importance of engineering, and quality in the creation of software process and its cycle will use bibliographic bibliographic research methodology.

Keywords: software engineering software quality software process.

1 Introdução

A engenharia de software é a responsável pela criação de softwares de qualidade que visa atender as necessidades do cliente, desenvolvendo produtos confiáveis, economicamente viável através de um conjunto de estrutura de dados e algoritmos na forma de procedimento, funções e módulos, desta forma desenvolvendo arquitetura de software que deverá ser executada em sistemas computacionais. Antes de chegar até o consumidor é feito através das software house um planejamento estimativa de projeto, análise de requisitos do software, projeto de estrutura de teste e manutenção. O planejamento e estimativa de projeto é feito para se ter uma noção de como o software será construído para satisfazer o cliente. Análise de requisitos é feita para revisar e implementar se houver falta de algum elemento, enfim observar se há erros. Projeto de estrutura de teste e manutenção é feito o projeto de estrutura de software como citado acima através de (algoritmos e estrutura de dados) e o teste e manutenção é feito para ver se o software funcionou de acordo com projetos feitos e com requisitos solicitado. Ao decorrer deste artigo através de estudos, pesquisas e entendimento demonstrar a importância da engenharia de software nas software houses para se ter um sistema de qualidade.

2 DESENVOLVIMENTO:

2.1 Software

Para Filho (2011), o software é uma entidade que se encontra em quase constante estado de mudança. Essas mudanças ocorrem por necessidade de corrigir erros existentes no software e/ou de adicionar novos recursos e funcionalidades. Igualmente, os sistemas computacionais (isto é, aqueles que têm software como um de seus principais elementos) também sofrem mudanças frequentemente.

Verzello (1984) classifica o *software* em três tipos, sendo: **Software de sistema** - são programas utilizados para controlar e coordenar o *software*. **Software de linguagens** - são programas que traduzem programas escritos em linguagens mais ou menos semelhantes a língua inglesa transformando-a em binária

que é a linguagem utilizada pelos sistemas computacional. Além disso, os programas escritos para ajudar os desenvolvedores a escrever seus programas e a manter os programas já escritos a salvo, em bancos de dados especiais. **Software de aplicação** - são programas escritos para resolução de problemas comerciais ou prestação de outros serviços de processamento de dados aos usuários.

2.2 Engenharia

Segundo Rezende (2005), Engenharia é a arte das construções com base no conhecimento científico e empírico onde o científico é a arte adequada ao atendimento das necessidades humanas. E o empírico está relacionado as experiências práticas.

3 CONCEITO DE ENGENHARIA DE SOFTWARE

Segundo Nogueira (2009) apud Martin (1991), engenharia de software é o estudo dos princípios e sua aplicação no desenvolvimento e manutenção de Sistemas de software.

Engenharia de software foi criada em 1969, por Bauer na primeira grande conferência dedicada ao assunto: O estabelecimento e uso de sólidos princípios de engenharia para que se possa estabelecer economicamente um software que seja confiável e que funcione eficientemente em máquinas reais (PRESSMAN, 1995, p.31).

Tanto a engenharia de software como as técnicas estruturadas, são coleções de metodologias de software e ferramentas, é um conjunto de práticas para o desenvolvimento de soluções de software, ou seja, um roteiro que pode utilizar diversas técnicas. (NOGUEIRA, 2009).

A engenharia de software é uma forma de engenharia que aplica princípios da ciência da computação e matemática para alcançar soluções com melhor custo- benefício para o problema de software para IEEE engenharia de software trata-se da aplicação sistemática disciplinada quantificável para o desenvolvimento, operação e manutenção do software (IEEE,1990). (TONSING, 2008, p.64).

Hirama (2012, p.07), diz que “engenharia de software é a qualidade e produtividade com tecnologia”.

3.1 A Importância da Engenharia de Software

A engenharia de software abrange um conjunto de três elementos fundamentais: **métodos**, **ferramentas** e **procedimentos** que possibilita ao gerente o controle do processo de desenvolvimento do software e oferece ao profissional uma base para a construção de software de alta qualidade produtiva. **Método:** proporcionam os detalhes de "como fazer" para construir o software muitas vezes utilizada de forma gráfica ou orientada a linguagem especial. **Ferramentas:** proporcionam apoio automatizado ou semi-automatizado aos métodos. **Procedimentos:** Os procedimentos da engenharia de software constituem o elo de ligação que mantém juntos os métodos e as ferramentas e possibilita o desenvolvimento racional e oportuno do software de computador. Definindo a sequência em que os métodos serão aplicados, que exigindo que sejam entregues (documentos, relatórios, formulários.) os controles que ajudam a assegurar a qualidade e a coordenar as mudanças, e os marcos de referencia que possibilitam aos gerentes de software avaliar o progresso.(PRESSMAN, 1995).

Um software derivado da engenharia obtém um produto confiável, funcionando de maneira eficiente e economicamente viável, o mesmo deve funcionar de acordo com as necessidades do usuário. O software se caracteriza por um conjunto de componentes abstratos (estrutura de dados e algoritmos) na forma de encapsulamento, procedimento, funções, módulos, objetos ou agentes interconectados entre si, compondo uma arquitetura de software que deverá ser executada em sistemas computacionais.(TONSING,2008).

A engenharia de software foca no software como produto. Não entra neste escopo o softwares construídos apenas para passarem o tempo dos programadores (FILHO, 2009).

3.2 Importância da Qualidade do Software

A qualidade do software é importante porque é a responsável pela criação de programas de computador, estrutura de dados e documentação que servem para efetivar o método, processo ou controle lógico necessário (PRESSMAN, 1995).

Se não tiver um software de alta qualidade seu projeto não terá nenhum valor, segundo Hetzel, a qualidade é não tangível, para Juran é a satisfação das necessidades do consumidor e a adequação ao uso. Crosby acredita que a qualidade é a conformidade às especificações que prevenir não conformidade é mais barato que corrigir ou refazer o trabalho, Chrissis, Konrad e Shrum disseram que a qualidade é a capacidade de um conjunto de características inerentes de um produto, componente ou processo para atender os requisitos de um cliente (HIRAMA, 2011, p.140).

“Para muitos engenheiros de software a qualidade do processo de software é tão importante quanto à qualidade de um produto” (NOGUEIRA, 2009, p.6).

Segundo Nogueira (2009) apud Bartié (2002), Qualidade de software é um processo sistemático que focaliza todas as etapas e partes produzidos com o objetivo de garantir a conformidade de processos e produtos, prevenindo e eliminando os defeitos.

Pressman (1995), apud McCall e Cavano (1978), assegura que as métricas para qualidade de software é: Transição do Produto, Revisão do Produto e Operação do Produto.

3.3 A Importância da Engenharia de Software nas Softwares House

Os métodos de engenharia de software demonstram os detalhes de como fazer para construir o software. Eles envolvem um amplo conjunto de tarefas que incluem: planejamento, estimativa de projeto, análise de requisito de software e de sistemas, projeto de estrutura de teste e manutenção (PRESSMAN, 1995).

Segundo Ambler (1998), um processo, ainda, pode ser definido como uma série de ações na qual uma ou mais entradas são utilizadas para produzir uma ou mais saídas.

Segundo Cordeiro (2000), os processos usados para desenvolver um projeto de software têm a maior importância na qualidade do software produzido e na produtividade alcançada pelo projeto.

Purri (2006), os desenvolvedores de software científico possuem grande carência acerca de um processo de desenvolvimento específico para a concepção de seus sistemas computacionais.

3.4 Paradigmas da Engenharia de Software

Um paradigma é um conjunto de atividades relacionadas a um padrão de resolução de problemas de um determinado gênero de programas e linguagens linguagem de programação (TUCKER, 2008)

os processos de software têm o seu principal enfoque nos processos técnicos e gerenciais relacionados ao desenvolvimento de software cujo objetivo é: planejar, monitorar, gerenciar, executar, controlar e melhorar atividades relacionadas a software. Dessa forma, a correta definição do conjunto de processos das atividades da empresas, agrupados, documentados e institucionalizados subsidiam a geração de produtos com altos níveis de qualidade (WEBER, 2002 apud-HAUCK2004 , p.33).

Sistemas de software, como quaisquer sistemas complexos, evoluem ao longo do tempo. Seus requisitos, muitas vezes, são difíceis de serem estabelecidos ou mudam com frequência ao longo do desenvolvimento (PRESSMAN, 2011).

O desenvolvimento de um software, qualquer que seja o modelo compreende três grandes fases: requisitos, projeto/desenvolvimento e implantação manutenção. (TONSING, 2008).

Requisitos: Os requisitos de um sistema de software incluem descrições das funções que o sistema deve prover e das restrições que devem ser satisfeitas. Em outras palavras, os requisitos definem o que o sistema deve fazer e as circunstâncias sob as quais deve operar (SOMMERVILLE, 2011).

Projeto/desenvolvimento: nessa fase o analista e programadores irão construir o software propriamente dito, o analista de sistemas sabendo o que deve ser desenvolvido e como os processos e dados são utilizados, após observar o levantamento de requisitos, faz as especificações técnicas detalhando a solução que ele criou para atender usuário/cliente, em seguida os programadores codificam os programas em alguma linguagem de programação e os testam na sua individualidade e coletivamente depois que todo o sistema foi testado, é liberado para o uso (TONSING, 2008).

Implantação/manutenção: a implantação requer algum cuidado além do técnico, porém é nesta fase que se manifestarão força de resistência ao software, caso elas existam. A manutenção do software permanecerá durante toda a vida útil, ocorre motivada por três fatores: a correção de algum problema existente no software, sua adaptação decorrente de novas exigências (de legislação por exemplo) e algum melhoramento funcional que seja incorporado ao software. (TONSING, 2008).

4 CICLOS DE VIDA DO SOFTWARE

De acordo com Tonsing (2008), Existem diversos tipos de modelos de documentação na engenharia, alguns sem nenhum projeto ou documentação outros de forma sequencial, como exemplo o modelo balbúrdia (sem projeto/documentação) e o modelo cascata (onde o software é desenvolvido de forma sequencial) veja cada um deles entre outros a seguir: **Modelo Balbúrdia:** o software é construído sem nenhum projeto ou documentação.

Modelo cascata: Apresenta uma visão de alto nível, que acontece durante o desenvolvimento e sugere aos desenvolvedores a sequência de eventos que eles devem esperar encontrar. (PFLEEGER, 2004)

Modelo Incremental: é um melhoramento do modelo anterior. Tem o objetivo de oferecer um produto operacional a cada incremento, e é individualmente útil quando não há mão de obra ou solução técnica disponível para a prática completa, dentro do prazo comercial de entrega colocado para o projeto. (OLIVEIRA, 2012).

Prototipação: é um processo que possibilita que o desenvolvedor crie um modelo do software que deve ser construído para uma prévia avaliação tanto do cliente quanto do desenvolvedor (JUNIOR, 2007).

Modelo espiral: foi criado por Dr. Berry Boehm em 1988, onde ele analisou o processo de desenvolvimento de software a partir do risco envolvido, sugerindo que um modelo em espiral poderia combinar as atividades de desenvolvimento com o gerenciamento de risco, de modo a minimizar e controlar os riscos, e a medida que o desenvolvimento do software avança, percorre-se a espiral do centro para fora. (PFLEEGER, 2004).

Modelos mistos e característica genérica: para essa criação podem-se pegar modelos existentes e gerar derivados. ou pode fazer uma combinação dos modelos existentes. (TONSING, 2008).

Scrum: interligado com as características que definem a qualidade de um software relacionando com as boas práticas da engenharia funcionam como um framework(abstração que une códigos provendo funcionalidade genérica) para gerenciamento de projetos que é capaz de oferecer melhora na qualidade de software, com a finalidade de entregar ao cliente o seu produto de acordo com as suas necessidades.(GRZEBIELUKA, 2016)

4.1 Qualidade do Software:

Construindo um software com qualidade cria –se real possibilidade de extrair de um sistema, informações relevantes que venham não só para contribuir com a decisão mas para ser um fator de excelência empresarial permitindo novos negócios, permanência e sobrevivência num mercado atuante. Para tanto é de suma importância identificar, analisar os riscos que ameaçam o sucesso do projeto bem como gerenciar para que se possa alcançar os objetivos empresariais. Todo fabricante visa qualidade dos produtos que produzem da mesma maneira os engenheiros de software procuram métodos que assegurem que seus produtos são de qualidade e utilidade aceitáveis, logo bons engenheiros de software devem sempre utilizar uma estratégia para a produção de software de qualidade. (NOGUEIRA, 2009, p.6).

Segundo Pfleeger (2004) apud Garvin (1984), Lawrence (2004) escreveu diferentes formas de como as pessoas percebem a qualidade. Ele descreve a partir de cinco perspectivas diferentes sendo elas: A visão transcendental, onde a qualidade é algo que podemos reconhecer, mas não podemos definir. A visão do usuário, onde a qualidade é a adequação ao propósito pretendido.

A visão do produto, onde a qualidade está relacionada às características inerentes ao produto. (PFLEEGER, 2004).

Qualidade palavra derivada do latim (qualitate), significa aquilo que caracteriza uma pessoa ou que a distingue das outras. (FERREIRA, 2004)

Vasconcelos (2006), complementa afirmando que em geral a qualidade esta relacionada com, normalização e melhoria de processo, medições, padrões e verificações.

4.2 Garantia de Qualidade

Para comparar a qualidade de um produto com a de outro, é necessário que possamos medir a qualidade do software. Primeiro identificamos os aspectos particulares do sistema que contribuem para a qualidade do produto como um todo. Assim quando os usuários forem medir a qualidade do software eles avaliarão características externas, tais como a quantidade e o tipo de falhas. (PFLEEGER, 2004).

Krutchen (2001), define algumas práticas que devem ser adotadas, visando a melhoria da qualidade dos produtos gerados, bem como um maior controle sobre o seu desenvolvimento. São elas: **Desenvolvimento iterativo:** permite um desenvolvimento mais realista, pois divide o projeto em ciclos menores, conhecidos como iterações, sendo que em cada uma delas são executadas todas as fases do desenvolvimento, gerando assim uma versão executável. **Gerência dos requisitos:** a característica mutável dos requisitos é certa. Esta prática busca gerenciar estas mudanças, para que a qualidade do produto não seja alterada por este motivo. **Uso de arquiteturas baseadas em componente:** um sistema deve ser desenvolvido utilizando componentes já prontos, certificados e testados, o que caracteriza reuso de código. Esta prática oferece grandes vantagens como a manutenção localizada. **Modelo visual do software:** a adoção de modelos gráficos como uma notação padronizada permite um entendimento geral da equipe, o que permite uma maior clareza na construção do sistema. **Verificação constante da qualidade do software:** com o controle constante de qualidade, a detecção precoce de uma falha acontece de forma natural, o que impede a sua propagação para as próximas etapas do desenvolvimento. **Controle de Mudanças do Software:** visa controlar e padronizar o desenvolvimento, para garantir a integridade na atualização dos produtos de trabalho.

Visando a qualidade no processo de desenvolvimento de software a gestão dos riscos tem o foco de tratar as incertezas inerentes aos projetos de software, pois muitos fatores que envolvem a tecnologia, pessoas e processos, se conflitam e podem determinar ou não o produto do software. (NOGUEIRA, 2009, p.6).

Qualidade significa ter alto valor agregado nos produtos sem considerar os níveis de produtividade e por outro lado ter alta produtividade significa ter muitos produtos sem considerar os níveis de qualidade. Assim um produto com qualidade é mais caro do que aquele obtido somente com produtividade, atualmente, bons níveis de qualidade e produtividade são essenciais em empresas competitivas. (HIRAMA, 2012 p.142).

4.3 Processo de Software:

Um processo é uma sequência de passos realizados para um dado propósito colocado de maneira mais simples, processo é aquilo que você faz. É aquilo que as pessoas fazem usando método procedimento, ferramentas e equipamentos, para transformar em matéria prima (entradas) em produtos (saída) que tenha valor para o cliente. (NOGUEIRA, 2009,p.2).

Os processos de desenvolvimento surgiram como parte da engenharia de software. Seu principal objetivo é assegurar a qualidade no desenvolvimento de sistemas em todas as suas etapas, para isso eles servem de guia para a execução de atividades, definem quais produtos serão desenvolvidos e quando, coordenam as mudanças necessárias, e auxiliam o acompanhamento do progresso do desenvolvimento do software (PIMENTEL, 2007).

Gonçalvez (2000), afirma que processo é um conjunto de atividades que recebe dados de entrada que realiza um processamento que adiciona valor a esses dados e produz uma saída de valor a um cliente específico.

“É uma série de etapas que envolvem atividades, restrições e recursos para alcançar a saída desejada”. (PFLEEGER, 2004, p.36)

O desenvolvimento de software de alta qualidade não é uma tarefa trivial desde o surgimento da indústria de software. No início os sistemas eram desenvolvidos sem a utilização de métodos sistemáticos e praticamente sem nenhuma documentação (ad hoc). O sucesso e a qualidade dos softwares dependiam diretamente dos profissionais envolvidos em seu desenvolvimento (FARIAS, 2006, p.29)

4.4 Normas para Qualidade de Software

A série ISO-9000 foi elaborada em 1979 pelo ISO Technical Committee 176 (ISO-TC-176) esse foi o primeiro dos comitês do grupo ISO a tratar dos assuntos da gestão e garantia de qualidade, a primeira versão só foi oficialmente publicada em 1987. (ANTONIONE/ROSA,1995)

Durante a década de 1950 e 1960, a qualidade era uma responsabilidade exclusiva do programador. Padrões de garantia de qualidade para o software foram introduzidos no desenvolvimento de software sob contrato militar durante a década de 1970 e espalharam-se rapidamente para o desenvolvimento de software no mundo comercial. (PRESSMAN, 2009, p.733).

segundo KOSCIANSK(2006) A qualidade do software depende da especificação dos requisitos, a não observação de erros e solicitações resulta em um software de qualidade baixa.

Segundo Antonioni (1995), a sigla ISO origina-se de International Organization for Standardization. Esse é o nome de um grupo internacional de normalização localizado em Genebra, Suíça. Essa organização foi fundada em 23 de fevereiro de 1947 não possui ligações com órgãos governamentais.

A série 9000 é composta por cinco documentos numerados sequencialmente: ISO-9000, ISO-9001, ISO-9002, ISO-9003 E ISO-9004. ISO-9000: normas de Gestão da Qualidade e Garantia da Qualidade diretriz para Seleção (da norma mais adequada ao caso da empresa) e uso. ISO-9001: Sistemas da qualidade-modelo para garantia da qualidade em projeto desenvolvimento, produção instalação e assistência técnica. ISO-9002: sistemas da qualidade-modelo para garantia da qualidade em produção e instalação. ISO-9003: sistemas de qualidade-modelo para garantia da qualidade em inspeção e estados finais. ISO-9004: gestão da qualidade e elementos do sistema da qualidade-diretrizes. (ANTONIONI, 1995).

A avaliação da qualidade do software pode ser feita atrás dos atributos internos, externos ou os atributos de qualidade em uso, com o objetivo de satisfazer a qualidade de software e obter um resultado solicitado num contexto particular. Esses atributos impostos pelo usuário não devem ser totalmente definidos, por não ser de real necessidade. **Usabilidade (utilizabilidade):** capacidade do produto de software de ser compreendido, aprendido, operado e atraente ao usuário, quando usado sob condições específicas. Sub-características: Inteligibilidade, Apreensibilidade e Operacionalidade. (NBR ISO/IEC 9126-1).

Manutenibilidade:É a capacidade do produto de software de ser modificado. As modificações podem incluir correções, melhorias ou adaptações do software devido a mudanças no ambiente e nos seus requisitos ou especificações funcionais; (MENDES, 2011).

4.5 Ciclo de Processo de Software

Um processo é um conjunto de procedimentos, organizados de modo que nos permita construir produtos que satisfaçam a uma serie de objetivos e padrões, processo que envolve a elaboração de um produto é chamado de ciclo de vida.O

software em desenvolvimento pode ser chamado ciclo de vida do software, pois ele descreve a vida do produto de software desde a concepção até a implementação. (PFLEEGER, 2004).

A estrutura do processo permite examinar, entender, controlar e aprimorar as atividades que o compõem. Os processos são importantes também porque nos permite capturar experiências e passar adiante (PFLEEGER,2004, p.37)

Durante o ciclo de vida de software são executados vários processos sendo que cada um contribui para atingir os objetivos de um estágio do ciclo (NOGUEIRA, 2009).

A melhoria dos processos de software pode ser implementada de diferentes maneiras. Ela pode ocorrer por meio da padronização de processos, pois é a primeira etapa essencial na introdução de novos métodos e novas técnicas de engenharia de software (NOGUEIRA, 2009, p.5).

"O desenvolvimento de software pode ser comparado com a construção de uma casa, necessita de arquiteto, empreiteiro, pedreiro, eletricitas, encanadores etc". (HIRAMA, 2012, p.6).

CONCLUSÃO

Os elementos fundamentais da engenharia que dão detalhes de como desenvolver o software que atenda as necessidades dos clientes tornando-o de alta qualidade produtiva são: os métodos, as ferramentas e os procedimentos sendo que os métodos auxiliam como fazer, as ferramentas fornecem o apoio automatizado aos métodos e os procedimentos são os elos de ligação entre o método e as ferramentas.

A engenharia de Software é composta por duas bases de conhecimento: o Empírico que são os conhecimentos práticos e o científico que são as teorias unindo os dois para melhor adaptação do software de acordo com a proposta da empresa.

O software se caracteriza por um conjunto de componentes abstratos estrutura de dados e algoritmos na forma de funções e módulos, procedimentos interconectados entre si, ou seja, foca no software como um produto adaptado às necessidades do cliente.

A qualidade do software é importante para que haja conformidade com o que foi proposto, ela focaliza todas as etapas e partes produzidas com o objetivo de garantir a eficácia e eficiência do produto prevenindo-o contra erros.

A engenharia de software nas software house é de suma importância pois deve-se observar as necessidades e perspectivas dos clientes começando sempre pelo planejamento, levantamento de requisitos, prototipação, desenvolvimento, testes e manutenção do software

REFERÊNCIAS

ANTONIONI José; ROSA, Newton Braga, **Qualidade em Software Manual de aplicação ISO-9000**-SãoPaulo Makron Books,1995.

AMBLER, S. W. **An Introduction to Process Patterns**.1998.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **Engenharia de software - Qualidade de produto**. NBR ISO/IEC 9126-1. Rio de Janeiro, 2003.

CORDEIRO, M. A., **Foco no processo**. 2000. Companhia de Informática no Paraná – CELEPAR.

FARIAS, T. M.M. **“Aplicação de Padrões ao Processo de Desenvolvimento de Software RUP.”** Trabalho de Conclusão de Curso - Universidade de Pernambuco, 2006

FERREIRA, Aurélio B. Holanda. **Aurélio Dicionário da Língua portuguesa**. 6ed. Rio de Janeiro: Positivo, 2004.

FILHO Paula, W.P. **Engenharia de software: fundamentos, métodos e padrões**. 3. ed. Rio de Janeiro: LTC,2009.

FILHO, Antônio Mendes da Silva ***Desenvolvimento de Software requer Processo e Gestão.ISSN:1519-6186**

GONÇALVEZ, J. E. L. **“As empresas são grandes coleções de processos.”** Revista de Administração de Empresas, Jan/Mar 2000: 6-19

GRZEBIELUKA Rafael de Figueiredo, ISSN:2236 6717,**Data de Publicação:** 28/01/2016 **Nº Certificado:** 20160128.005208 **Edição da Revista:** 000080 **Volume da Revista:**01

HAUCK, J. C. R. **“Desenvolvimento de um Sistema de Software para Gerência de Manuais de Processos de Software em Micro e Pequenas Empresas.”** Trabalho de Conclusão de Curso 2004.

HIRAMA, Kehi, **Engenharia de Software Qualidade e Produtividade do Software**, 2º ed. Elsevier, Rio de Janeiro, 2011.

KOSCIANSKI, André; Soares, Michael dos Santos. **Qualidade de software- Aprenda as metodologias e técnicas mais modernas para desenvolvimento de software**. 2.ed. São Paulo: Novatec Editora,2007.

KRUCHTEN, P., **What is the Rational Unified Process?**. Copyright Rational Software 2001content/RationalEdge/jan01/WhatIsTheRationalUnifiedProcessJan01

JUNIOR Manoel Pereira **Belo Horizonte Centro Federal de Educação Tecnológica de Minas Gerais Diretoria de Pesquisa e Pós-Graduação 2007**

MENDES, Telma Carolina, **Estudo da garantia de qualidade de software utilizando a metodologia de desenvolvimento de sistemas criada por uma instituição financeira**, FATEC, São Paulo, 2011. 62 p.

NOGUEIRA, Marcelo **Engenharia de software um framework para a gestão de riscos em projetos de software**. Rio de Janeiro: Editora Ciência moderna LTda.2009

OLIVEIRA, Leila Ribeiro de. **Implementação de processos: o uso de técnicas de estimativas de projetos de software para estimar processos de negócio**. Universidade FUMEC. Belo Horizonte, 2012.

PRESSMAN, R.S., **Engenharia de Software: Uma Abordagem Profissional**, 7ª Edição, McGraw-Hill, 2011.

PRESSMAN, Roger S. **Engenharia de Software**, Ciência moderna, São Paulo, 1995.

PURRI, M. C. M. S. **Estudo e Propostas Iniciais para a Definição de um Processo de Desenvolvimento para Software Científico**. Dissertação de Mestrado. Centro Federal de Educação Tecnológica de Minas Gerais – CEFET-MG. Belo Horizonte, 2006.

PURRI, M. C. M. S.; PEREIRA Jr, M.,; MOITA, G. F. **PESC – Processo de Desenvolvimento Específico para Software Científico: Propostas Iniciais. XXVII CILAMCE – Iberian Latin American Congress on Computational Methods in Engineering**. Setembro 2006. Belém-PA.

PIMENTEL, A. R. **“Uma Abordagem para Projeto de Software Orientado a Objetos Baseado na Teoria de Axiomático”**. Tese de Doutorado – Universidade Tecnológica Federal do Paraná, Maio 2007.

PFLEEGER Shari Lawrence, **Engenharia de Software Teoria e Prática** 2ªedição São Paulo,2004.

REZENDE, Denis Alcides **Engenharia de Software e Sistemas de Informação**/.”3ªed. rev e ampl.”Rio de Janeiro:Brasport,2005.

SOMMERVILLE, I., **Engenharia de Software**, 9ª Edição. São Paulo: Pearson Prentice Hall, 2011

TONSIG, Sergio Luiz, **Engenharia de software analise e projeto de sistemas** 2ªedicao, Rio de Janeiro, Ciência Moderna, 2008

TUCKER, Allen B./NOONAN Robert E, **Linguagens de programação: princípios e paradigmas**, 2ªed.sao Paulo: Mc GRAW-HILL,2008.

VASCONCELOS, Alexandre Marcos Lins de. [et. al]. **Engenharia de Software Para Software Livre1**. Lavras:UFLA,2006

VERZELLO, Robert J.; REUTTER III, John **Processamento de dados**. São Paulo : McGraw-Hill, 1984.

WEBER, S. “**Um Estudo de Caso em uma Microempresa de Software para o Desenvolvimento de um Modelo de Processo de Software.**” Universidade Federal de Florianópolis - UFSC, 2002.