

GABRIEL MATHEUS OLIVEIRA - 5203
JOÃO PEDRO PEREIRA - 5199

IMPLEMENTAÇÃO DOS MODELOS DE QUALIDADE DE
SOFTWARE PARA A MELHORA EM MICRO E PEQUENAS
EMPRESAS

Trabalho de projeto de pesquisa para o final do semestre elaborado para a matéria de Tratamento e análise de dados
Orientador: Leandro Furtado

RIO PARANAÍBA

2017

Resumo

Este trabalho demonstra que pequenas empresas necessitam adotar um modelo de qualidade de software, devido a seu investimento ser menor, um que segue as normas técnicas de padrão de qualidade e que as ajude a crescer. Alguns modelos são mais conhecidos, como o CMM e spice, ambos seguem normas da ISO/IEC, ao serem aplicados em redes de mercados, nos gera algumas suposições de resultados que possam acontecer em outros tipos de empresas de tamanho semelhante e independente de seus fornecimentos para a população.

Palavras-chaves: Qualidade, software, modelos, maturidade, CMM, MPS.br, ISO, IEC, spice, CMMI.

Abstract

This work demonstrates that small companies need to adopt a software quality model, due to their smaller investment, one that follows the technical norms of quality standard and that helps them to grow. Some models are better known, such as CMM and spice, both follow ISO / IEC standards, when applied in networks of markets, it generates some assumptions of results that can happen in other types of companies of similar size and independent of their supplies for the population.

Key-words: quality, software, models, maturity, CMM, MPS.br, ISO, IEC, spice, CMMI

Lista de ilustrações

Figura 1 – Visão geral dos processos ISO/IEC 12207	12
Figura 2 – Níveis de maturidade CMM	14
Figura 3 – Avaliação de processos de software SPICE	14

Lista de tabelas

Tabela 1 – Cronograma	17
Tabela 2 – Cronograma futuro	17

Lista de abreviaturas e siglas

ABNT	Associação Brasileira de Normas Técnicas
CMM	Capability Maturity Model(modelo de maturidade capacitiva)
ISO	International Organization for Standardization(Organização Internacional para Padronização)
IEC	International Electrotechnical Commission(Comissão Eletrotécnica Internacional)
MPS.br	Melhoria de Processos do Software Brasileiro

Sumário

1	Introdução	8
2	Objetivos da pesquisa	9
2.1	Objetivo Geral	9
2.2	Objetivos Específicos	9
3	Justificativa	10
4	Referencial Teórico	11
4.1	Engenharia de software	11
4.2	Processo de software	11
4.3	Qualidade de software	11
4.3.1	Norma ISO/IEC 12207	11
4.3.2	SEI SW-CMM	12
4.3.3	SPICE	13
4.3.4	MPS.BR	13
5	Trabalhos Relacionados	15
6	Metodologia e Cronograma	17
7	Resultados Esperados	18
	Referências	19

1 Introdução

Ao longo do tempo, grandes, médias e pequenas empresas buscam cada vez mais se modernizar tecnologicamente, não só como um tipo de investimento para melhorias estruturais e visuais, existe também a necessidade de atualizar para não se perderem no mercado em relação as outras empresas, a qualificação dos funcionários, divisão de tarefas, sendo que algumas passam a ser automatizadas pelos softwares desenvolvidos ou adquiridos e melhoria no desempenho do processo empresarial. Adquirir softwares, para as pequenas empresas pode ser um passo fundamental para seu crescimento, entretanto, elas sofrem mais ao comprar o produto, pois estes podem trazer prejuízo por não apresentarem um bom fator de qualidade.

Garantir que o software funcione corretamente não é uma tarefa fácil, levando em conta o tamanho e a complexidade do projeto e é o teste o responsável pela garantia de qualidade resultando em uma grande importância no desenvolvimento de um software. Através dele é possível encontrar falhas nos sistemas, antes que o mesmo seja entregue ao cliente. O teste pode ser feito de forma manual ou automatizada apresentando diferentes tipos de técnicas e níveis de teste de software, fornecendo o suporte necessário para garantir testes adequados para cada fase do desenvolvimento de um software, pois cada etapa pode gerar diferentes falhas. Por isso fica clara a necessidade de se conhecer melhor os teste de software para que sejam desenvolvidos softwares mais seguros, confiáveis e de com muito mais qualidade.

A pesquisa sobre a qualidade de software vem sendo constantemente desenvolvida, várias técnicas são criadas e antigas são aperfeiçoadas visando sempre a melhoria dos produtos. Pela falta de capital e recursos, empresas de pequeno porte buscam produtos confiáveis, que foram passados por todos os procedimentos, tipos de análise de qualidade e que estejam em seu orçamento. Contudo, alguns produtos não passam por essa inspeção, o quanto um produto desqualificado pode prejudicar financeira e estruturalmente essas empresas e quais análises devem ser feitas para saber se são adequadas as exigências que à empresa oferece sobre o produto, são temas abordados no texto.

2 Objetivos da pesquisa

2.1 Objetivo Geral

Esclarecer, do ponto de vista pequeno-empresarial, a importância ao adquirirem produtos que possuem seus padrões baseados em testes desenvolvidos pela qualidade de software.

2.2 Objetivos Específicos

- Validar se somente as normas técnicas de análise sobre a qualidade de software são justificáveis para seguir seus modelos e aplica-los em empresas
- Comparar as vantagens de pequenas empresas que possuem produtos passados pelos métodos com produtos que não foram.
- Questionar sobre o tempo de vida de softwares nessas empresas e o que fazer quando decair sua qualidade.

3 Justificativa

Conforme a necessidade das empresas, a qualidade de software é sempre indispensável. Como alguns exemplos, um almoxarifado de uma empresa, o CFOP de uma mercadoria, e principalmente os dois tributos mais complexos do Brasil, o PIS e o COFINS, tendo que ser preciso ao relacionar no seu programa de banco de dados a entrada e saída de materiais. Caso contrario, poderia ocorrer diferenças para a produção, controle e vendas do produto final e em casos mais graves, ocorre em problemas de notas fiscais eletrônicas.

O custo do defeito é progressivo, ou seja, encontrar o defeito na fase de engenharia de requisitos custa 1 enquanto encontrar o defeito durante a fase de uso custa 100 vezes mais, então utilizar o teste, reduz custo e não aumenta (Pressman, 2015).

Como grandes empresas têm mais condições se algo der errado, procuramos entender como um software com defeito em menores organizações as prejudica, se isso acarreta somente perdas financeiras ou pode também causar seus desligamentos, por um motivo que da para se prevenir apenas seguindo alguns cuidados. Nenhuma empresa começa com grande alcance em vendas e produtos, por isso é importante ter certeza de estar criando suas mercadorias com confiança e qualidade.

4 Referencial Teórico

4.1 Engenharia de software

Aplicação dos princípios científicos da engenharia, a partir de um padrão qualitativa fundamentada e organizadamente visando obter softwares com maior qualidade, usabilidade, portabilidade e eficiência nas máquinas. É dividida em camadas para acrescentar propriedades em seu produto, através de gerenciamento dos processos, métodos e das ferramentas necessárias para manter a qualidade no software. Se gasta mais para manter o software que para desenvolvê-lo, nos sistemas de longa vida útil, os custos de manutenção podem maiores que os custos de seu desenvolvimento.

4.2 Processo de software

Metodologias usadas para o desenvolvimento do software, a fim de possibilitar a equipe desenvolvedora de produzir seu produto dentro de um prazo e com qualidade na entrega. Entre esses processos compreendem algumas atividades mais importantes, como: comunicação com os interessados, planejamento da jornada, esboço ou modelagem, construção, revisão e emprego avaliativo ao cliente.

4.3 Qualidade de software

É de conhecimento geral que em qualquer produto adquirido, o consumidor exige que este tenha uma qualidade, garantia acessibilidade e utilidade para ele. Em questão de softwares ocorrem os mesmo, alguns produtos fogem dos interesses de quem os adquiriu, para isso, foram realizadas técnicas para se criar um nível base a ponto satisfazer o consumidor. Cada região se adapta de acordo com a demanda. Para o Brasil, por exemplo, a SOFTEX (Sociedade para Promoção da Excelência do Software Brasileiro) diz já existir alguns modelos padrões dessa melhoria nos processos de softwares, que seguem as normas implicadas pela ISO (normas técnicas mundiais) e IEC (normas internacionais relacionadas a áreas eletrônicas), para os desenvolvedores seguirem e manter existente a qualidade que satisfaça os usuários dos softwares.

4.3.1 Norma ISO/IEC 12207

A norma técnica que foi elaborada em junho de 1989, tem como objetivo principal estabelecer uma estrutura comum para os processos de ciclo de vida e de desenvolvimento de softwares. Afim de ajudar organizações a compreender todos os requisitos presentes

ao adquirirem, desenvolverem e fornecerem um software. Para isso é necessário que todos envolvidos no desenvolvimento de software usem um framework comum.

Seu padrão envolve ao todo 25 processos, 95 atividades e 325 tarefas. Uma tarefa é uma ação com entradas e saídas. Pode ser um requisito, recomendação, ou permissão. Uma atividade é um conjunto de tarefas. Um processo é um conjunto de atividades relacionadas. A Norma define dezessete processos do ciclo de vida de um software e os organiza em três classes: processos fundamentais, processos de apoio e processos organizacionais. Cada classe contém os processos definidos e os possíveis usuários, como mostra a figura 1.

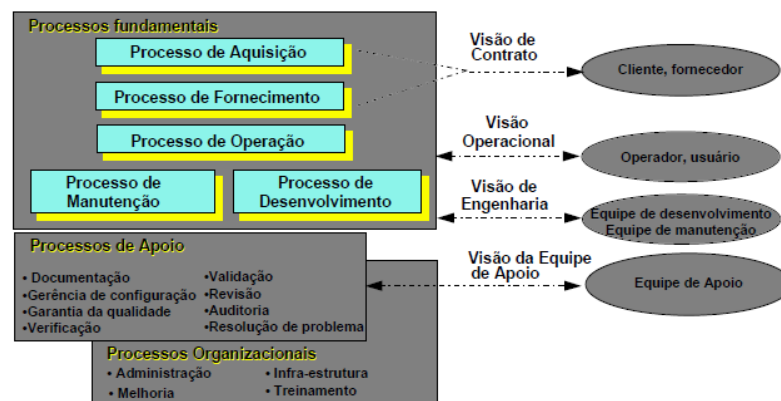


Figura 1 – Visão geral dos processos

ISO/IEC 12207

4.3.2 SEI SW-CMM

O modelo Capability Maturity Model propõe a avaliação da capacidade e maturidade de uma organização e indica caminhos para a melhoria. Dispõe de cinco níveis de maturidade definidos e baseados em princípios de qualidade de produto propostos por Walter Shewart, W. Edwards Deming, Joseph Juran e Philip Crosby, altamente reconhecidos como mestres da qualidade na área de gerenciamento.

A escala dos níveis de maturidade estão apresentados na figura 2.

A primeira etapa de avaliação para o CMM é o treinamento da equipe de avaliação, a segunda etapa é o planejamento da avaliação, a terceira etapa é a execução da avaliação, por último vem a reportagem dos resultados, que ocorre no último dia de avaliação.

Cada nível tem seus devidos processos, menos o inicial, como exemplos de processos, no nível gerenciado, tem gerenciamento de requisitos, acompanhamento e controle de projeto, entre outros, no nível definido temos programas de treinamento, definição do processo organizacional, e assim em diante. Ao todo cada nível tem 18 áreas de processos.

4.3.3 SPICE

Esse projeto foi lançado em 1993, após o comitê de Engenharia de Software da ISO aprovar a realização de estudos para analisar as necessidades e requisitos de um padrão para processos de software. Ele foi criado também, com objetivo de gerar normas para avaliar processos, buscando sua melhoria e capacitação. Seu modelo de referência na verdade é uma estrutura de avaliação à partir de outros modelos como SW-CMM, Trillium e STD, e seu resultado virou uma nova ISO/IEC, a 15504.

O modelo possui um ciclo que em seu inteiror, os softwares são avaliados, para entender seus estados de processos, determinar uma adequação e gerenciar as mudanças necessárias. Ocorre conforme a demonstração na figura 3.

4.3.4 MPS.BR

A MPS.BR foi criada para atender perfis de vários tipos de empresas, sempre padronizando os processos e melhorando a qualidade dos softwares. Esse modelo se baseia nas seguintes normas: ISO/IEC 15504 e 12207 e no modelo CMMI, que é um avanço do CMM, com um número maior de processos em cada nível de maturidade.

A melhoria contínua do processo é estabelecida por meio de sua avaliação quantitativa e da implantação planejada e controlada de tecnologias e ideias. Projetos são realizados para a absorção de novas tecnologias. Seus vários níveis avaliativos por integrantes e membros da sessão executiva são validados e considerados no final da avaliação. Esse modelo então, foi criado exatamente para inibir altos investimentos, pois os custos de modelos como o CMMI são inacessíveis para algumas corporações o que acabava dificultando a venda dos produtos de melhoria de software.

A necessidade de uma reestruturação organizacional de empresas de pequeno e médio porte com o passar do tempo é certa, sendo assim, o modelo MPS.BR dispõe de mais níveis de maturidade e sugere a implantação do nível de maturidade G (parcialmente gerenciado), que é composto pelos processos mais críticos de gerência. Ao implementar os processos deste nível a empresa pode focar os esforços de melhoria em estabelecer mecanismos mais adequados para o planejamento de projetos, monitoração e controle, e para gerenciar requisitos ao longo do ciclo de vida do produto. Esse modelo, se resume entre um dos os mais adequados para a realização das mudanças. O modelo é aceito por se basear em normas reconhecidas internacionalmente, desde a definição de cargos até a adoção dita do projeto, saindo mais em conta para a seleção e implementação empresarial.

CMM - NÍVEIS DE MATURIDADE

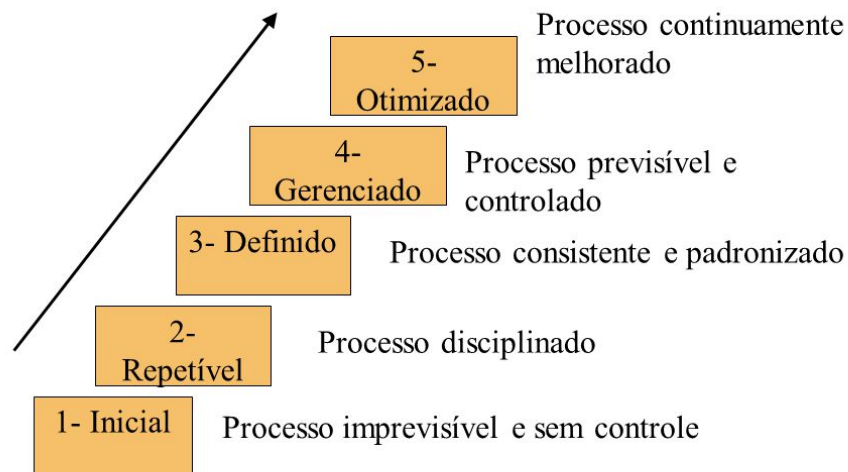


Figura 2 – Níveis de maturidade

SW-CMM

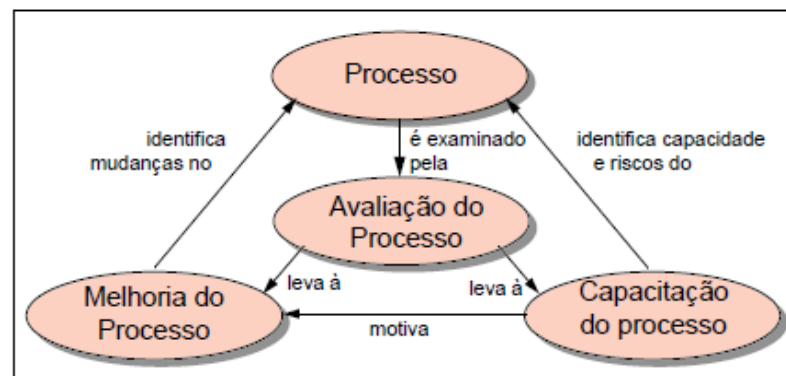


Figura 3 – Avaliação de processos de software

SPICE

5 Trabalhos Relacionados

O artigo Aplicando uma Metodologia Baseada em Evidência na Definição de Novas Tecnologias de Software , escrito em 2006 pelo Professor de Engenharia de Software Guilherme Horta Travassos, ilustra como a utilização de uma metodologia baseada em evidência pode auxiliar a minimizar falhas em tecnologias de software com inspeção de documentos arquiteturais.

Avaliação da Qualidade de Software com Base em Modelos UML, escrito em 2009, foi a dissertação apresentada como requisito parcial para obtenção do título de Mestre da estudante Isela Macía Bertrán que propõe técnicas de análise da qualidade capazes de detectar problemas de design.

Em [Gesing, José Carlos (17/06/2011)] se tem um trabalho que apresenta um estudo sobre as normas ISO/IEC 12207 e ISO/IEC 15504 e sobre o modelo para melhorar o processo de software MPS.BR com destaque no nível G, aplicado em uma microempresa, assim como um breve comparativo ao modelo CMM. Deve-se lembrar, que a MPS.BR foi criada para atender perfis de vários tipos de empresas como já dito, padronizando os processos e melhorando a qualidade dos softwares. A MPS.BR foi criada exatamente para inibir altos investimentos, adotando o modelo, pois os custos de modelos como o CMMI são inacessíveis para algumas corporações o que acabava dificultando a venda dos produtos de melhoria de software.

Mafra, Sômulo Nogueira, et al. “Aplicando uma Metodologia Baseada em Evidência na Definição de Novas Tecnologias de Software.” propos um artigo que ilustra como a utilização de uma metodologia baseada em evidência pode auxiliar a minimizar falhas em tecnologias de software recém-definidas no contexto industrial através da técnica de leitura de requisitos de software e de uma abordagem para inspeção de documentos arquiteturais. A metodologia experimental para introdução de tecnologias de software na indústria é composta por quatro etapas, que contemplam a avaliação da tecnologia desde sua definição até sua transferência para a indústria. As quatro etapas são representadas por: estudos de viabilidade, estudos de observação, ciclo de vida, e estudo de caso na indústria.No trabalho em questão, pode se identificar o processo de mudanças na adoção de métodos que são introduzidos em empresas. Também se viu a evidencia referente a minimização de falhas por meio de aplicação dos metodos através de estudo sobre os mesmos.

A PhD Isela Macía Bertrán fez um dissertação sobre "Avaliação da Qualidade de Software com Base em Modelos UML.". Esta dissertação divulga técnicas de análise da qualidade capazes de detectar problemas de design, como um conjunto de estratégias de detecção para identificar, em modelos UML, problemas de design específicos e recorrentes na literatura e a utilização do modelo da qualidade QMOOD para avaliar design de

software a partir de seus diagramas de classes. Em primeiro lugar, o modelo UML é a linguagem padrão de elaboração de estruturas de projeto de software. Ela permite a visualização do trabalho em si, não a maneira de como administrar seus projetos, o modelo também possibilitou a criação de diversos tipos de diagramas estruturais. Já no modelo QMOOD, este por sua vez, tem a função de avaliar os desings de softwares, para a automatização e melhoria na qualidade de software a partir da análise de seus diagramas de classes.

Lima, Felipe Augusto. “DESENVOLVIMENTO ENXUTO DE SOFTWARE.2014, p.50 ” Universidade Tecnológica Federal do Paraná, Pós-Graduação em Gestão de Negócios, Curitiba, 16 Mar. 2014. O artigo estuda a aplicabilidade da produção enxuta voltada ao desenvolvimento de software, bem como o grau de conhecimento/informação deste sistema pelos profissionais que atuam no ramo de desenvolvimento de software.

Avaliação da Validade Externa da Técnica de Análise Diferencial para Detecção de Envelhecimento de Software: Um Estudo Confirmatório com Replicação, escrito por Guilherme Otávio de Sena em 2017, é a dissertação para obtenção do título de mestre em Ciência da Computação que, resumidamente, fala sobre o envelhecimento de software baseada em uma análise diferencial de software. O texto apresentado é relevante, pois softwares envelhecidos aplicados em alguma organização, são prejudiciais a mesma, mesmo sendo inspecionados pelos padrões de qualidade.

O pesquisador Elias Delgobo Junior, em 2015, desenvolveu um instrumento de coleta de dados com o objetivo de identificar o uso de modelos de avaliação e de melhoria de processos em desenvolvedoras de software em Curitiba. O impacto do uso dos modelos é positivo, no primeiro momento o processo de melhoria se torna caro, com gastos com treinamento e na mudança cultural da empresa, em uma segunda etapa, depois da consolidação do uso dos modelos, a empresa passa a ter economia, com menos erros e gastos desnecessários. O autor propõe essa pesquisa e mudanças dos métodos dos modelos que ele cita, como por exemplo o MPS.Br já dito em outros trabalhos, assegurando que os clientes utilizaram o software e responderam as perguntas, gerando assim, conhecimento sobre o que não é entendido para tentar mudar se possível e realmente necessário, para agradar o consumidor.

6 Metodologia e Cronograma

1. Aplicação de dois modelos diferentes, que se enquadram nas normas ISO/IEC, o CMMI e MPS.Br, em um software de controle do estoque de alguma rede de mercado alimentício que possui uma filial para realizar uma comparação entre a aplicação dos modelos, afim de definir qual traz mais vantagens a essa pequena rede comercial.
2. Realizar outra comparação, com uma diferente rede de mercado da cidade, analisando as vantagens da rede que teve seu software passado pelo modelo definido, tanto organizacional quanto em produção e renda.
3. Depois de um período, verificar a necessidade de substituição do produto, caso seu tempo de vida estiver prejudicando à qualidade, porém, aplicando novamente o modelo na organização.

O cronograma esperado para a verificação dos métodos é mostrado na Tabela:

Tabela 1 – Cronograma do Projeto(Meses)

Atividade	1	2	3	4	5	6	7	8	9	10	11	12
Revisão Bibliográfica						•	•					
Aplic. do método 1							•	•	•			
Aplic. do método 2								•	•	•		
Resultados esperados										•	•	
Entrega de resultados												•

Após a entrega dos resultados, para um próximo periodo do ano, uma nova verificação sugestiva no caso de se desejar adquirir um novo software.

Tabela 2 – Cronograma do Projeto em Meses caso se adquira um novo produto

Atividade	1	2	3	4	5	6	7	8	9	10	11	12
Teste do novo software				•	•							
Aplicação do modelo					•	•	•					
Comparação entre os produtos							•	•	•			
Resultados aplicados										•		

Fonte: João e Gabriel

7 Resultados Esperados

Depois de diferir o modelo considerado mais vantajoso pelas empresas alimentícias dos testes, a aplicação deste nos softwares de controle dos estoques nos faz crer que não haja perda de informação, que mantenha ou melhore a segurança na empresa, facilite a monitoração dos empregados e faça o software ser mais interativo com os usuários.

Para o próximo passo, a implementação em outras redes, nos gera a certeza que a qualidade é absolutamente necessária para micro e pequenas empresas. Os resultados apareceram de forma rápida, pois uma organização que tem um software de controle não inteligível, precário e duvidoso, possivelmente está propenso à erros que podem acarretar em percas para a empresa. Essas percas não são interessantes, pois lucrativamente um investimento em um produto duvidoso, sai muito mais caro que a inserção do modelo escolhido para qualidade, embora as vezes, nem serem necessárias trocas, apenas redefinições das estruturas do programa.

Até mesmo um bom produto, corre o risco de ficar ultrapassado tecnologicamente, pois nos tempos de hoje, a tecnologia sempre se desenvolve. Para que não se gere prejuízos, todo novo software adquirido futuramente por esses mercados, deve passar também pelos modelos avaliativos, entretanto, o atual gera uma garantia que até que se veja a necessidade de troca.

O teste realizado em pequenas empresas alimentícias nos trás uma questão: Os modelos aplicados para qualidade do software só valem para esse tipo de empresa? A resposta é não, qualquer ramo empresarial pode se beneficiar do uso, desde que o problema em seus utilitários se encaixem no benefício trazido pela revisão de qualidade de software. Para exemplificar, uma empresa de produção textil tem um software que a partir de buscas em seu estoque, sugere produtos relacionados ao consumidor. Caso esse software de algum bug, essa empresa pode perder vendas, diminuir seu marketing e o seu alcance. Com o modelo implementado, esses bug's não serão um problema aos vendedores, pois testes de possíveis erros são feitos e corrigidos, então sim, se pode aplicar em diversos ramos de micro e pequenas empresas distintas.

Referências

- ANACLETO, A. et al. Método e modelo de avaliação para melhoria de processos de software em micro e pequenas empresas. Florianópolis, SC, 2004.
- BARTIÉ, A. **Garantia da qualidade de software**. [S.l.]: Gulf Professional Publishing, 2002.
- BR, M. Mps. br-melhoria de processo do software brasileiro. 2011.
- HECK, A. C. Dynamic cmm como alternativa para melhoria de qualidade de processos em micro e pequenas empresas de software. **Monografia apresentada na Universidade Tecnológica Federal do Paraná, Medianeira**, 2012.
- KOSCIANSKI, A. **Qualidade de Software - Segunda Edição: Aprenda as metodologias e técnicas mais modernas para o desenvolvimento de software**. [S.l.]: <https://books.google.com.br/books?id=O9aWoUq6L88C>, 2007.
- MACHADO, C. F. Definindo processos do ciclo de vida de software usando a norma nbr iso/iec 12207 e suas ementas 1 e 2. **Lavras: UFLA/FAEPE**, 2006.
- ROCHA, A. Influência da maturidade da função sistema de informação na abordagem à engenharia de requisitos. 2000.
- SENA GUILHERME OTÁVIO, I. y. Avaliação da validade externa da técnica de análise diferencial para detecção de envelhecimento de software: Um estudo confirmatório com replicação.
- TONINI, A. C. et al. Contribuição dos modelos de qualidade e maturidade na melhoria dos processos de software. **Produção**, Associação Brasileira de Engenharia de Produção, v. 18, n. 2, p. 275–286, 2008.
- TSUKUMO, A. N. Qualidade de software: Visões de produto e processo de software. **CTI - Fundação Centro Tecnológico para Informática**, 2007.
- VIANA, R. D. Uma análise sobre a importância da utilização do cmm e mps-br na garantia da qualidade do produto. http://www.ietec.com.br/site/techoje/categoria/detalhe_artigo/1262., 2012.
- WEBER, K. C. et al. Modelo de referência para melhoria de processo de software: uma abordagem brasileira. In: **XXX Conferencia Latinoamericana de Informatica (CLEI2004), Sesión**. [S.l.: s.n.], 2004. v. 13, p. 20–10.
- WEBER, S.; HAUCK, J. C. R.; WANGENHEIM, C. V. Estabelecendo processos de software em micro e pequenas empresas. **Simpósio Brasileiro de Qualidade de Software (SBQS), Porto Alegre**, 2005.

(WEBER et al., 2004) (TONINI et al., 2008) (BARTIÉ, 2002) (KOSCIANSKI, 2007) (ROCHA, 2000) (WEBER; HAUCK; WANGENHEIM, 2005) (ANACLETO et al., 2004) (BR, 2011) (MACHADO, 2006) (TSUKUMO, 2007) (VIANA, 2012) (HECK, 2012) (SENA GUILHERME OTÁVIO,)