



GOVERNO DO
ESTADO DO CEARÁ

*Secretaria da Ciência, Tecnologia
e Educação Superior*



UNIVERSIDADE ESTADUAL
VALE DO ACARAÚ

Engenharia de Software

1º Bloco

NEaDUVA

Núcleo de Educação a Distância
Universidade Estadual Vale do Acaraú
SOBRAL - 2013

Capítulo 2 – Processo: Uma visão Genérica

Segundo Howard Baetjer, Jr.(1998, apud Pressman, 2010) processo é um diálogo no qual o conhecimento que, deve se transformar em software é reunido e incorporado ao software. Ele fornece interações múltiplas entre os projetistas, usuários e ferramentas em desenvolvimento. Onde essas ferramentas são os meios de comunicações do diálogo.

Um processo de software define a abordagem que é adotada quando o software é elaborado, mas a Engenharia de Software também inclui tecnologias que constituem um processo – métodos técnicos e ferramentas automatizadas.

Engenharia de Software – Uma tecnologia de em camadas

A engenharia de software é realizada por pessoas criativas, com amplos conhecimentos, que devem adaptar um processo de software maduro apropriado para os produtos que elas constroem e para as demandas do seu mercado.

O IEEE [IEE93] desenvolveu uma definição mais abrangente: Engenharia de Software: aplicação de uma abordagem sistemática, disciplinada e quantificável, para o desenvolvimento, operação e manutenção do software; isto é, a aplicação da engenharia ao software. (2) O estudo de abordagens com as de. Porém o que é sistemático, disciplinado e quantificável para um equipe de software pode ser cansativo para outra, pois precisam de agilidade.

Como visto na figura abaixo, Pressman (2010) fala que a Engenharia de software pode ser dividida em camadas (Processo, Métodos e Ferramentas) onde todas essas camadas tem como base o foco na qualidade.



Processos:

O processo é o adesivo que mantém unidas as camadas de tecnologias e permite o desenvolvimento racional e oportuno de softwares de computador. Os processos formam a base para o controle gerencial de projetos e estabelece o contexto no qual os métodos são aplicados, os produtos de trabalhos são produzidos, a qualidade é assegurada e as modificações são adequadamente geridas.

Métodos:

Os métodos fornecem a técnica de “como fazer” para construir softwares, eles abrangem tarefas como comunicação, modelagem de projeto, construção de programas, testes e manutenção.

Ferramentas:

As ferramentas fornecem apoio automatizado ou semi-automatizado para os processos e para os métodos. Ferramentas são criadas de modo que a informação criada por um ferramenta possa ser usada por outra.

Arcabouço de Processo

Um Arcabouço de Processo, através de um pequeno número de atividades, estabelece o alicerce para um processo de software, e possui um conjunto de atividades que são aplicáveis em todo o processo de software, listados abaixo:

Comunicação: Diálogo com o clientes e outros interessados para levantar requisitos e outras atividades.

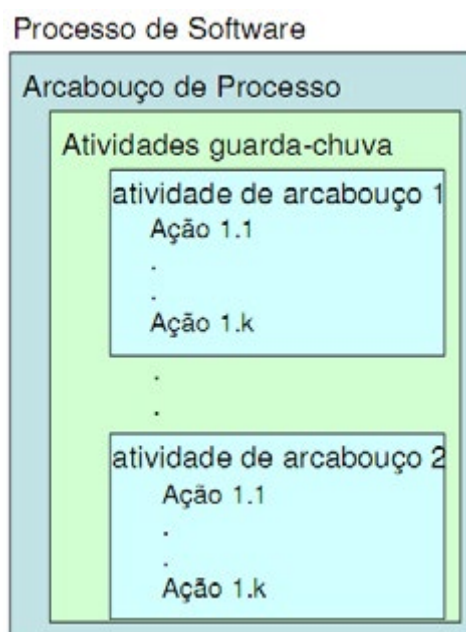
Planejamento: Plano de Trabalho, cálculo de riscos, de recursos, e cronograma de trabalho.

Modelagem: Modelos que ajudam a entender os requisitos do software.

Construção: Geração do código e os teste necessário para procurar erros.

Implantação: Entrega do software ao cliente.

Dependendo do Desenvolvimento os detalhes do processo poderão ser diferentes, mas as atividades do arcabouço permanecem as mesmas.



O arcabouço referente a figura acima é complementado por várias atividades guarda-chuva, que incluem:

- Acompanhamento e controle de projeto;
- Gestão de risco;
- Garantia de qualidade;
- Revisões técnicas formais;
- Medição (satisfação do usuário);
- Configuração;
- Reusabilidade;
- Preparação e produção do produto do trabalho.

Todos os modelos de processo podem ser caracterizados dentro do arcabouço de processo. A aplicação inteligente de qualquer modelo de processo de software deve reconhecer que a adaptação (ao problema, ao projeto, à equipe e à cultura organizacional) é essencial para o sucesso.

Modelos de Processos

Modelos de Processos é a atividade de representação de processos de uma empresa, de modo que o processo atual pode ser analisado e melhorado. Modelagem de processos de negócio é normalmente realizado por analistas de negócios e gestores que estão buscando melhorar a eficiência do processo e da qualidade.

Os modelos diferem fundamentalmente:

- No fluxo geral de atividades e tarefas
- No grau em que as tarefas de trabalho são definidas dentro de cada atividade do arcabouço
- Na maneira como as atividades de garantia de qualidade são aplicadas
- No grau geral de detalhes e rigor
- No grau de envolvimento dos clientes
- No nível de autonomia da equipe de desenvolvimento.

Os modelos prescritivos de processo, enfatizam a definição, identificação e aplicação detalhada e atividades e tarefas onde seu objetivo é melhorar a qualidade do sistema para tornar os projetos mais gerenciáveis, as datas de entrega e os custos mais previsíveis e para guiar equipes de engenheiros de software. Se aplicados dogmaticamente e sem adaptação, eles podem aumentar a burocracia e criar dificuldades para desenvolvedores e clientes.

Já os modelos de processo ágil, enfatizam a agilidade do projeto, a manobrabilidade e a adaptabilidade, possuem uma abordagem mais informal mas não menos efetiva. Ambas as filosofias têm como objetivo comum criar softwares de alta qualidade que satisfaçam às necessidades do

cliente.

CMMI (Capability Maturity Model Integration)

O SEI (Software Engineering Institute) desenvolveu um abrangente metamodelo de processo baseado em um conjunto de capacidades de engenharia de software que devem estar presentes à medida que as empresas alcançam diferentes níveis de capacidade e maturidade de processo.

O CMMI representa um metamodelo de processo de dois modos diferentes: como um modelo contínuo e como um modelo em estágios. Cada área de processo é avaliada formalmente com base em metas e práticas específicas, e é classificada de acordo com os seguintes níveis de capacitação:

Nível 0: Incompleto – A área de processo não atinge todas as metas e objetivos definidos pelo CMMI.

Nível 1: Realizado – Todas as metas específicas da área de processo foram satisfeitas.

Nível 2: Gerido – Todos os critérios do nível 1 foram satisfeitos. Além disso, todo trabalho associado à área de processo está de acordo com a política definida pela organização.

Nível 3: Definido – Todos os critérios do nível 2 foram alcançados. Além disso, o processo é “feito sob medida para o conjunto-padrão de processos da organização”.

Nível 4: Quantitativamente gerido – Todos os critérios do nível 3 foram alcançados. Além disso, a área de processo é controlada e aperfeiçoada usando medições e avaliação quantitativa.

Nível 5: Otimizado – Todas as capacidades do nível 4 foram alcançadas. Além disso, a área de processo é adaptada e otimizada usando meios quantitativos.

Muito da crise de software é auto-infligida, como quando um executivo-chefe de informática diz: “Eu preferia ter isso errado do que atrasado. Podemos sempre corrigir isso depois”. (Mark Paulk).

O CMMI define cada área de processo em termos de “metas específicas” e das “práticas específicas”. As metas específicas estabelecem as características que devem existir se as atividades determinadas por uma área de processo tiverem de ser efetivas. As práticas específicas refinam uma meta num conjunto de atividades relacionadas ao processo. O **planejamento de projeto** é uma das oito áreas do projeto definidas pelo CMMI para a categoria “gestão de projeto”.

Meta Específica 1 – Estabelecer Estimativas

Meta Específica 2 – Desenvolver um plano de projeto

Meta Específica 3 – Obter comprometimento com o plano

Além de metas e práticas específicas, o CMMI também define um conjunto de cinco metas genéricas e praticas relacionadas para cada área de processo.

Meta Genérica 1 – Alcançar metas específicas

Meta Genérica 2 – Institucionalizar um processo gerido

Meta Genérica 3 – Institucionalizar um processo definido

Meta Genérica 4 – Institucionalizar um processo quantitativo gerido

Meta Genérica 5 – Institucionalizar um processo otimizado

PADRÕES DE PROCESSO

Um padrão de processo de software pode ser visto como um conjunto de atividades, métodos, ferramentas e práticas que são utilizadas para construir um produto de software.

Um processo de software pode ser definido como uma coleção de padrões, que devem ser consideradas as seguintes informações: atividades a serem realizadas, recursos necessários, artefatos requeridos e produzidos, procedimentos adotados e o modelo de ciclo de vida utilizado. Pela combinação de padrões, uma equipe de software pode construir um processo que melhor satisfaça às necessidades de um projeto. Pelas quais, se as atividades operarem corretamente e de acordo com os padrões requeridos, o resultado desejado é produzido, no qual, é um software de alta qualidade e baixo custo.

Humphrey (1990, apud Macoratti 2013) define as seguintes razões para a definição de um processo de software padrão:

- Redução dos problemas relacionados a treinamento, revisões e suporte à ferramentas;
- As experiências adquiridas nos projetos são incorporadas ao processo padrão e contribuem para melhorias em todos os processos definidos;
- Economia de tempo e esforço na definição de novos processos adequados a projetos.

Em termos gerais, um padrão de processo nos fornece um gabarito, um método consistente para descrever uma característica importante do processo de software.

Ambler (1998, apud Pressman, 2010) propôs o seguinte gabarito para descrever um padrão de processo:

Nome do padrão. Ao padrão é dado um nome significativo que descreve sua função dentro do processo de software (por exemplo, comunicação com o cliente).

Intenção. O objetivo do padrão é descrito brevemente. Por exemplo, a intenção de comunicação com o cliente é “estabelecer um relacionamento colaborativo com o cliente esforçando-se para definir o escopo do projeto, os requisitos do negócio e outras restrições de projeto”.

Tipo. Ambler (1998, apud Pressman, 2010) sugere três tipo:

- *Padrões de tarefa:* definem uma ação de engenharia de software ou tarefa de trabalho. Exemplo, **levantamento de requisito**. Se uma empresa deseja estabelecer um contrato para o desenvolvimento de um projeto de software, os requisitos devem ser redigidos de modo que os diversos fornecedores possam apresentar propostas. O fornecedor precisa preparar uma definição de sistema para o cliente contendo mais detalhes, de modo que o cliente possa compreender e validar o que o software fará.
- *Padrões de estágio:* definem uma atividade de arcabouço (arcabouço é o alicerce ou esqueleto de um processo de software completo) para o processo, incorpora vários padrões de tarefas que são relevantes para o estágio. Exemplo, **comunicação**, que incorpora o padrão de tarefa levantamento de requisitos.
- *Padrões de fase:* definem a sequência de atividades de arcabouço que ocorrem no processo, mesmo que o fluxo global de atividades é de natureza iterativa. Exemplo, **prototipação**.

Contexto inicial. Antes da iniciação do padrão, pergunta-se (1) que atividades relacionadas à organização ou à equipe já ocorreram, (2) qual é o estado das entradas para o processo e (3) que informações de engenharia de software ou informações de projetos já existem. Exemplo, o padrão **planejamento**.

Problema. O problema a ser resolvido pelo padrão é descritivo. Exemplo, comunicação com o cliente não é estabelecida ou inadequada para o levantamento de informações.

Solução. A implementação do padrão é descrita. Esta seção discute como o estado inicial do processo é modificado em consequência da iniciação do padrão.

Contexto resultante. Ao término do padrão pergunta-se: (1) que atividades relacionadas à organização ou à equipe devem ter ocorrido, (2) qual é o estado de saída do processo, (3) que informações de engenharia de software ou informações de projeto foram desenvolvidas.

Padrões relacionados. Uma lista de todos os padrões de processos diretamente relacionados a este é fornecido.

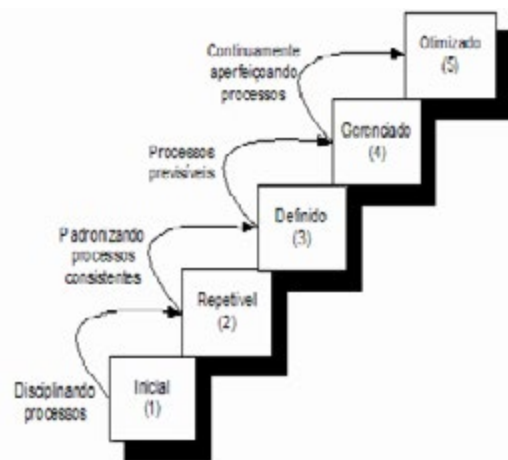
Usos conhecidos/Exemplos. São indicadas as instâncias específicas nas quais o padrão é aplicável. Exemplo, **comunicação**, que é obrigatória no início e ao longo de todo o projeto de software.

Avaliação de processo

Várias e diferentes abordagens de avaliação de processo de software têm sido propostas durante as últimas décadas.

O **SCAMPI (standard CMMI Assessment Method for Process Improvement, Método de Avaliação Normalizado do CMMI para Aperfeiçoamento de Processo)** fornece um modelo de processo de avaliação com cinco passos que incorpora a iniciação, o diagnóstico, o estabelecimento, a ação e o aprendizado. O método SCAMPI usa o CMMI da SEI como base para avaliação.

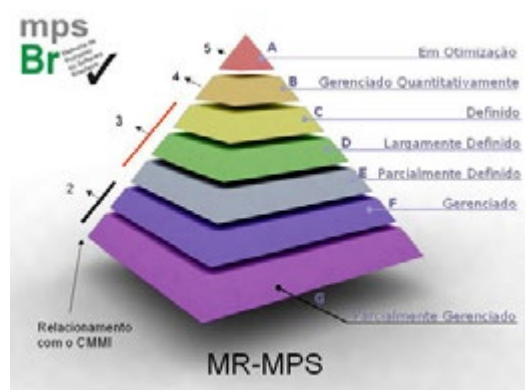
A **CBA IPI (CMM-Based Appraisal for internal Process Improvement, Avaliação do Processo de Aperfeiçoamento Interno Baseado no CMM)** fornece uma técnica de diagnóstico para avaliar a maturidade relativa de uma organização de software usando o CMM da SEI como base para avaliação. Níveis de maturidade do CMMI:



A **norma SPICE (ISO/IEC 15504)** define um conjunto de requisitos para avaliação de processo de software. A intenção da norma é assistir às organizações no desenvolvimento de uma avaliação objetiva da eficácia de qualquer processo definido de software.

A **ISO 9001:2000** para software é uma norma genérica que se aplica a qualquer organização que deseje aperfeiçoar a qualidade global dos produtos, sistemas ou serviços que ela fornece. Assim a norma é diretamente aplicável a organizações e empresas de software.

MPS.BR - Melhoria de Processo do Software Brasileiro é baseado nas normas ISO/IEC 12207 e ISO/IEC 15504 e na realidade do mercado brasileiro, bem como é compatível com o CMMI. O MPS.BR apresenta 7 níveis de maturidade (o que é um diferencial em relação aos outros padrões de processo).



MODELO DE PROCESSO PESSOAL E DE EQUIPE

O melhor processo de software é aquele que se aproxima do pessoal que fará o serviço. O ideal seria: cada engenheiro de software criaria um processo que melhor satisfizesse às suas necessidades e, ao mesmo tempo, às necessidades mais globais da equipe. Alternativamente: a própria equipe poderia criar o seu próprio processo e, ao mesmo tempo satisfazer às necessidades mais restritas dos indivíduos e às necessidades mais amplas da organização.

PSP (Personal Software Process – Processo Pessoal de Software)

Derivado do CMM, o autor deste processo é o mesmo do CMM (Humphrey).

PSP foi criado para auxiliar o desenvolvedor a estimar e planejar suas tarefas, acompanhar sua performance em relação ao planejado e melhorar a qualidade dos produtos produzidos, além de enfatizar a mediação pessoal tanto do produto do trabalho que é produzido quanto da qualidade resultante do produto do trabalho.

Torna o profissional responsável pelo planejamento do projeto e dá poder ao profissional para controlar a qualidade de todos os produtos do trabalho de software que são desenvolvidos. O modelo de processo PSP define cinco atividades de arcabouço que são:

Planejamento. Isola requerimentos, desenvolve estimativas, toda a métrica é registrada em planilhas ou gabaritos e as tarefas de desenvolvimento são identificadas e um cronograma de projeto é criado.

Projeto de alto nível. São desenvolvidas especificações externas para cada componente a ser construído e é criado um projeto dos componentes. Todos os itens são registrados e monitorados.

Revisão do projeto de alto nível. Métodos de verificação formal são aplicados para descobrir erros no projeto. A métrica é mantida para todas as tarefas e resultados importantes do trabalho.

Desenvolvimento. O projeto em nível de componente é refinado e revisado. O código é gerado, revisado, compilado e testado. A métrica é mantida para todas as tarefas e resultados importantes do trabalho.

Pós conclusão. Usando as medidas e a métrica coletadas, a efetividade do processo é determinada, e, através de suas diretrizes, pode-se modificar os processos de modo a aperfeiçoar sua efetividade.

Sugere práticas e métodos para que o próprio indivíduo consiga identificar e corrigir seus pontos fracos.

É uma sugestão para organizar e disciplinar os processos individuais e não diminuir nem restringe a capacidade criativa dos indivíduos.

PSP0 - Os profissionais de software aprendem a aplicar os formulários e roteiros do PSP aos seus trabalhos pessoais, medindo tempos e defeitos de desenvolvimento, defeitos estes injetados ou removidos.

PSP0.1 - Os profissionais de software registram no PMP os problemas, os tópicos importantes para discussão e argumentação e as idéias a serem usadas futuramente, aperfeiçoando assim os seus processos pessoais.

PSP1 - São elaborados planos de tamanho, recursos e tempos gastos no trabalho – Gerenciamento de Projetos.

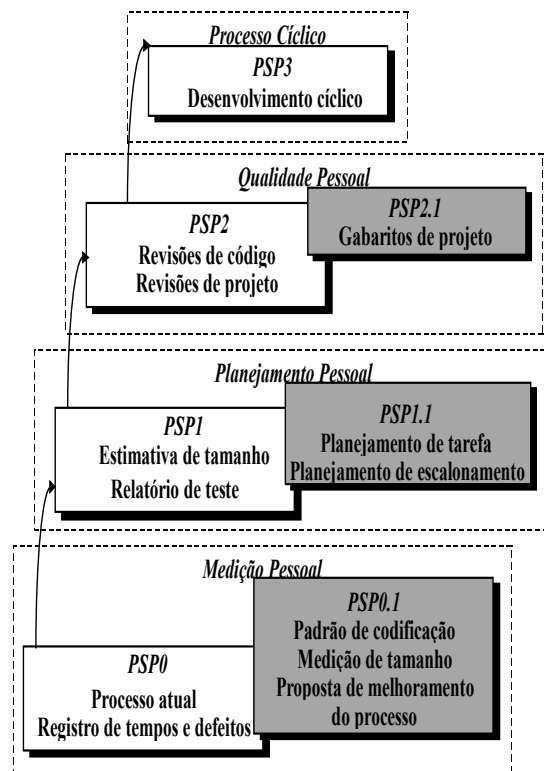
PSP1.1 - Adiciona o escalonamento e planejamento de tarefas.

PSP2 – Com os dados de defeitos reunidos previamente, os profissionais de software constroem e usam listas de verificação para revisão de projeto e código (checklists).

PSP2.1 - Introduz as técnicas de especificação de projeto e análise em adição à prevenção de defeitos, análise e comparação de processos.

PSP3 - Os métodos do PSP são ampliados para projetos maiores – Escalabilidade do Processo.

A Evolução do Processo PSP



TSP (Team Process Software – Processo de Equipe de Software)

Após a criação do PSP, Watts Humphrey propôs o TSP. TSP é uma estrutura para a melhoria quantitativa de processo de software que ajuda equipes a desenvolver produtos de software de modo eficaz.

Humphrey (1998, apud Pressman, 2010) define os seguintes objetivos para o TSP:

- Construir equipes de software autogeridas que planejem e monitorem seu trabalho, estabeleçam metas e possuam seus próprios processos e planos.
- Mostrar aos gerentes como acompanhar e motivar suas equipes.
- Acelerar o aperfeiçoamento do processo de software.
- Fornecer diretrizes de aperfeiçoamento para organizações de alta maturidade.

- Facilitar o ensino universitário das habilidades de equipe de nível industrial.

O TSP usa uma grande variedade de documentos, formulários e normas que servem para guiar os membros da equipe em seus trabalhos e reconhece que as melhores equipes de software são as autogeridas.

CMMI (Capability Maturity Model Integration)

É um modelo que auxilia as organizações na implementação de um conjunto de práticas de gestão e de desenvolvimento de sua força de trabalho, em vez de apenas alguns indivíduos. Fornece um guia de alto nível para o desenvolvimento do processo organizacional, não entrando em detalhes de como suas práticas devem ser implementadas.

Tecnologia de Processo

Segundo Feiler (1993, apud Link_5 2013) Tecnologia de Processo de Software surgiu no final da década de 1980, e, representou um importante passo em direção à melhoria da qualidade de software através de mecanismo que proporcionam o gerenciamento automatizado do desenvolvimento de software.

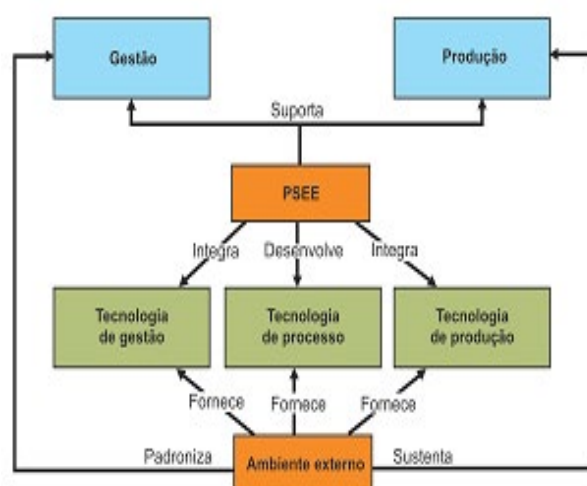


Figura 1: Tecnologia do Processo de desenvolvimento de Software

Os PSEEs (**Process Software Engineering Environments** ou **Processo de ambientes de engenharia de software**) Lima (1998, apud Link_5 2013) diz que, constituem um tipo especial de ambiente de desenvolvimento de software apoiando a definição rigorosa de processos de software e objetivando a automação da gerência do desenvolvimento. Tais ambientes geralmente proveem serviços para análise, simulação, execução e reutilização das definições de processos que, cooperam no aperfeiçoamento contínuo de processos, conforme mostra a Figura 1, sendo que alguns desses PSEEs utilizam recursos de inteligência artificial para apoiar o processo de desenvolvimento de software, Silva (2005, apud Link_5 2013).

A ferramenta de tecnologia de processo também pode ser usada para coordenar o uso de outras ferramentas de engenharia de software apoiadas por computador que sejam adequadas a uma tarefa de trabalho específica. Exemplo. Aplicação de workflow como suporte a processos de software em PSEE.

Workflow. As tecnologias de workflow (WF) têm sido utilizadas para modelar, melhorar e automatizar processos de negócios (PN). Um workflow pode ser visto como a automação total ou parcial de um processo de negócio e esta automação consiste em representar um workflow em um formato compreensível por uma máquina.

Um workflow é representado por um grafo dirigido onde os nodos são as suas atividades e as setas indicam a precedência de execução.



Desta forma, a tecnologia de workflow pode ser utilizada para dar apoio ao processo de software em um PSEE, pois ajuda os projetos de software a alcançar disciplina, controle e entendimento claro dos seus processos e atividades. Reis (1999, apud Link_5).

Sendo assim, as ferramentas de processo ajudam as organizações a analisar seu processo atual, organizar tarefas de trabalho, controlar e monitorar o progresso e gerenciar a qualidade técnica Negele (1999, apud Pressman 2010).

Um exemplo é a ferramenta de modelagem de processo, que são usadas para representar os elementos-chave de um processo de modo que ele possa ser melhor entendido. Modelagem é a parte central de todas as atividades que levam à implantação de um bom software.

Produto e Processo

Se o processo de desenvolvimento de um produto é ruim, sem dúvida o produto obtido é ruim. No entanto não se deve focar apenas no processo, o produto é também importante. O desenvolvedor de software deve sentir satisfação tanto na execução do processo de desenvolvimento como no produto final obtido. A qualidade de software é uma preocupação real e esforços têm sido realizados na busca pela qualidade dos processos envolvidos em seu desenvolvimento e manutenção. A mesma preocupação existe com o produto de software desenvolvido onde testes, normas e métodos são utilizados para verificar sua qualidade.

No entanto, o software continua com sua qualidade comprometida. Um processo de qualidade não garante um produto de qualidade. Percebe-se, neste ponto, uma lacuna nos esforços que vêm sendo realizados na busca pela qualidade de software. O processo, que irá resultar no produto de software, concentra seus esforços na busca pela qualidade do “modo de produção e manutenção” do software, enquanto a qualidade do produto de software é focada no produto final, através da constatação de sua qualidade por avaliações no software já acabado.

Conclusão

Baseado nesse contexto, foram abordados alguns conceitos que visam à melhoria da produtividade no desenvolvimento de software para a obtenção do aumento da qualidade do software produzido. Dependendo de sua aplicação, ambiente e objetivo, a utilização de um processo ou modelo específico pode ser vantajoso ou não. Cabe a cada organização avaliar o seu problema com cuidado e usar um modelo como guia para o desenvolvimento do seu próprio processo de desenvolvimento.

BIBLIOGRAFIA

PRESSMAN, R. S., Engenharia de software. Sexta edição.

[Link_1](#) - José Carlos Macoratti – O processo de software, acessado em 09 de janeiro de 2013.

[Link_2](#), acessado em 10 de janeiro de 2013.

[Link_3](#), acessado em 10 de janeiro de 2013.

[Link_4](#), acessado em 10 de janeiro de 2013.

[Link_5](#), acessado em 11 de janeiro de 2013.

[Link_6](#), acessado em 11 de janeiro de 2013.

[Link_7](#), acessado em 16 de janeiro de 2013.

[Link_8](#), acessado em 17 de janeiro de 2013.

[Link_9](#), acessado em 17 de janeiro de 2013.

[Link_10](#), acessado em 18.02.2013.