INPG – INSTITUTO NACIONAL DE PÓS-GRADUAÇÃO

CURSO DE PÓS-GRADUAÇÃO *LATO SENSU* EM
GERÊNCIA DE PROJETOS DE TECNOLOGIA DA INFORMAÇÃO

ANÁLISE DE COMPATIBILIDADE ENTRE CMMI NÍVEL 2 E PROGRAMAÇÃO EXTREMA

ANA PAULA DOS SANTOS MOTA

SÃO PAULO/SP, 2012

2012	**GERÊNCIA EM PROJETOS DE TECNOLOGIA DA INFORMAÇÃO**	**INPG**

INPG – INSTITUTO NACIONAL DE PÓS-GRADUAÇÃO

CURSO DE PÓS-GRADUAÇÃO *LATO SENSU* EM
GERÊNCIA DE PROJETOS DE TECNOLOGIA DA INFORMAÇÃO

ANÁLISE DE COMPATIBILIDADE ENTRE CMMI NÍVEL 2 E PROGRAMAÇÃO EXTREMA

ANA PAULA DOS SANTOS MOTA

SÃO PAULO/SP, 2012

ANA PAULA DOS SANTOS MOTA

ANÁLISE DE COMPATIBILIDADE ENTRE CMMI NÍVEL 2 E PROGRAMAÇÃO EXTREMA

Monografia apresentada ao INPG - Instituto Nacional de Pós-Graduação, como exigência para a obtenção do título de Especialista em Gerência de Projetos da Tecnologia da Informação.

Orientador: Prof. MsC. José Antônio Rosa

SÃO PAULO / SP

2012

INPG - Instituto Nacional de Pós-Graduação

ANÁLISE DE COMPATIBILIDADE ENTRE CMMI NÍVEL 2 E PROGRAMAÇÃO EXTREMA

Monografia apresentada pelo (a) aluno (a) **Ana Paula dos Santos Mota** ao curso de Especialização em Gerência de Projetos em Tecnologia da Informação em Data da entrega 03/02/2012.

_____

Orientador: Prof. MsC. José Antônio Rosa

Aprovada com nota

_____

Agradecimentos

Agradeço a Deus por tudo. A minha fé e os princípios em que acredito foram fundamentais durante a condução do meu caminho até aqui.

Aos meus pais e ao meu irmão pelo amor, incentivo e apoio. Pelos conselhos e orientações, que muitas vezes se mostraram sem sentido no primeiro momento, mas revelaram-se verdadeiros e certos com o passar do tempo.

Aos companheiros de trabalho da empresa Accenture pela oportunidade de crescimento e aprendizado com a implantação do CMMI nível 3. O conhecimento adquirido neste processo foi fundamental para a realização deste trabalho.

Aos colegas do curso de Gerência de Projetos de Tecnologia da Informação, por compartilhar o aprendizado durante estes dois anos. Em particular, aos amigos que estiveram mais próximos e fizeram parte do meu grupo de trabalho: Paulo Brigatti, Anderson Rodrigues, Fabíola Monteagudo, Alexandre Souza, David Alonso, Claudio Roberto e Fabio Meira.

Aos professores do Instituto de Matemática e Estatística da USP, Fabio Kon e Alfredo Goldman pela oportunidade de trabalhar com Programação Extrema durante minha graduação e mestrado.

Aos mestres do curso Gerência de Projetos de Tecnologia da Informação, pela paciência, auxílio, direcionamento e conhecimento compartilhado. Em especial, ao mestre Jakov Trofo Surjan que, além de ter ministrado várias disciplinas, também foi nosso coordenador de curso buscando sempre as melhores soluções para os nossos problemas.

Por fim, agradeço a todos que contribuíram direta ou indiretamente com este trabalho.

Muito obrigada!

“The world acquires value only through its extremes and
endures only through moderation;
Extremists make the world great,
the moderates give it stability.”
Paul Valery

RESUMO

A sociedade atual vivencia um rápido avanço tecnológico que a tornou cada vez mais dependente de sistemas e softwares. Estes sistemas precisam ser desenvolvidos rapidamente e de forma flexível para se adaptarem às mudanças constantes impostas por novos hardwares e novos negócios. Os fatores estratégicos, comerciais e humanos tornam a atividade de desenvolver software complexa e de alto risco.

Com a crescente demanda por software, surgiram vários modelos de desenvolvimento que visam aumentar a qualidade do produto entregue e mitigar os riscos envolvidos. Estes modelos de desenvolvimento dividem-se em dois grandes grupos: métodos conservadores ou tradicionais e métodos leves ou ágeis.

Os métodos tradicionais focam o processo de desenvolvimento. Um processo de desenvolvimento de qualidade produz produtos de qualidade. Esta é uma premissa há muito estabelecida pela área de manufatura e acreditar nela é uma tendência visível nos movimentos de qualidade da indústria e setores de serviços (Padrão ISO).

Já os métodos ágeis não valorizam o processo de desenvolvimento. Nesta metodologia, valorizam-se os indivíduos envolvidos no processo e a comunicação entre eles. Valoriza-se a colaboração efetiva com o cliente e atividades que trazem valor imediato na produção de software. A documentação é vista como uma atividade burocrática.

Devido a estas diferenças de valores, ambos os grupos tem sido visto como incompatíveis ou divergentes. O principal objetivo deste trabalho é estudar duas metodologias ou frameworks de processo de desenvolvimento de software que são consideradas opostas, e estabelecer algum nível de relação entre elas. As metodologias escolhidas foram o Capability Maturity Model Integration(CMMI), representando a vertente conservadora e clássica do processo de desenvolvimento, e Programação Extrema(XP), representando a nova tendência de metodologia leves ou ágeis.

Palavras-chave: CMMI, Programação Extrema, Metodologias Ágeis, Nível de Maturidade, Práticas

ABSTRACT

Nowadays, the society experiences a quickly technological advancement which has increased the dependence of systems and software. These systems must be developed in a quickly and flexible way to adapt to constant changes imposed by new hardware and new business. Strategic, commercial and human factors make software development a complex and high risk activity.

With the growing demand for software, there are various models of development aimed at increasing the quality of the delivered product and mitigate the risks involved. These development models can be classified into two main groups: traditional and conservative methods and light or agile methods.

Traditional methods focus on the development process. A quality process of development results into quality products. This assumption is a long established by the manufacturing area and believe it is a visible trend in the quality movement in industry and services sectors (Standard ISO).

Agile methods don´t give emphasis to the development process. In this methodology, the emphasis is on individuals involved in the process and on communication between them. Agile methods value the effective collaboration with clients and activities that adds immediate value in the production of software. Documentation is seen as a bureaucratic activity.

Because of these differences in values, both groups have been seen as incompatible or conflicting. The main objective of this work is to study two methodologies or frameworks for software development process that are considered opposite, and establish some level of relationship between them. The methods chosen were the Capability Maturity Model Integration (CMMI), representing the conservative and classical aspects of the development process, and Extreme Programming (XP), representing the new trend in light or agile methodology.

Palavras-chave: CMMI, Extreme Programming, Agile Methodologies, Maturity Levels, Pratices

Sumário

1. INTRODUÇÃO

1.1. Cenário

Os avanços tecnológicos tornaram a sociedade atual completamente dependente de computadores e sistemas. Todos os dias, novos equipamentos eletrônicos são produzidos exigindo a produção de softwares cada vez mais específicos e avançados. A crescente demanda por software torna o mercado cada vez mais competitivo, onde um dos diferenciais que uma organização pode ter é a maneira de produzi-lo. É preciso produzir software flexível, com qualidade e que consiga agregar valor ao negócio de forma rápida.

O desafio de produzir software de qualidade não é tão recente. A crise do software, no final de 1960, motivou a busca por teorias, métodos e ferramentas que possibilitassem a produção de software de qualidade com custo e prazos previsíveis. O termo *Engenharia de software* surgiu neste contexto. Segundo Sommervile[1], "o objetivo da Engenharia de Software é produzir produtos de software". A Engenharia de software trata de aspectos relacionados ao estabelecimento de processos, métodos, técnicas, ferramentas e ambientes de suporte ao desenvolvimento de software. Um dos objetivos da Engenharia de software é melhorar a qualidade dos produtos de software desenvolvidos[2].

Apesar de 40 anos de existência e muita evolução na discussão da qualidade de software, a Engenharia de Software ainda não foi capaz de resolver todos os problemas do processo de desenvolvimento. Os altos índices de projetos de software cancelados ou entregues com baixa qualidade ainda assustam. O Standish Group[3] realiza pesquisas de mercado e publicam alguns relatórios. O mais famoso deles é o CHAOS que traz dados interessantes sobre o desenvolvimento de software nas empresas. O gráfico abaixo mostra os resultados dos projetos de software para os anos de 1994, 1996, 1998 e 2000 publicados no CHAOS de 2001[4].

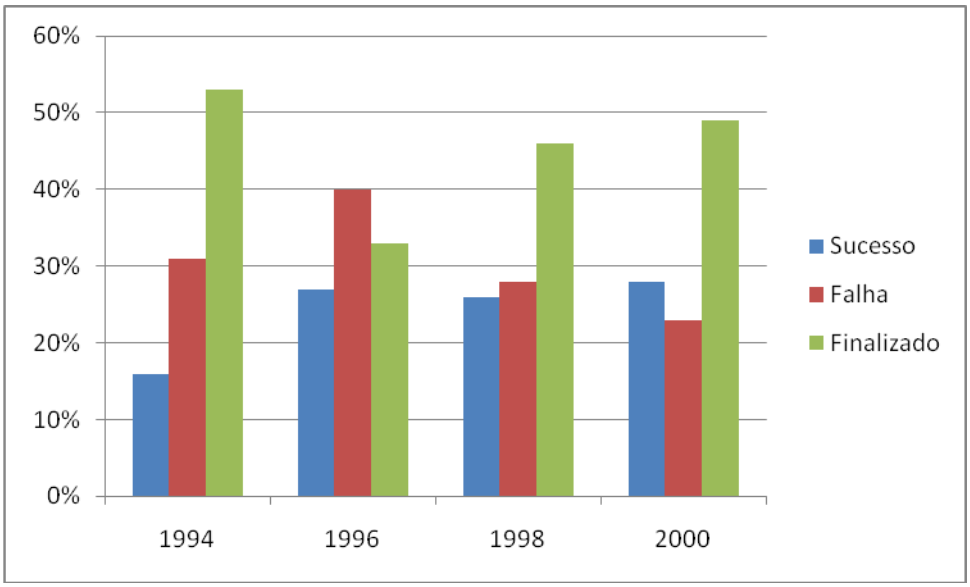

Figura 1 - Adaptado do CHAOS Report[4]

A classificação dos projetos em sucesso, falha ou finalizado é dada de acordo com os seguintes resultados:

- Sucesso: o projeto é entregue dentro do prazo e custo estimados, com todas as funcionalidades inicialmente especificadas.
- Falha: O projeto é cancelado antes de ser finalizado.
- Finalizado: O projeto é finalizado, porém com estouro de prazo ou custo estimado. Ou ainda, não possui todas as funcionalidades especificadas inicialmente.

A grande maioria dos projetos de software é classificada como finalizado (Figura 1 - Adaptado do CHAOS Report). O percentual de projetos de sucesso ainda é pequeno.

Mas o que fazer quando há falhas no processo de desenvolvimento de software? Quando as entregas atrasam constantemente e há sempre alterações de última hora? Quando as queixas do cliente são freqüentes? Quando o índice de retrabalho é alto e causa perdas na margem de lucro, frustrações para o time que, mesmo realizando muitas horas extras, tem a sensação de que não há ninguém no comando?

O que precisamos fazer é identificar quais são as causas do caos estabelecido e identificar pontos de melhoria. Mas para isso é preciso ter algum controle sobre o processo. É preciso definir metas a serem alcançadas para sair do caos.

Neste cenário, precisamos de um gerenciamento eficaz do processo. Uma premissa muito forte na área de qualidade de sistemas é que foco na qualidade do processo significa foco na qualidade do produto final. Esta é uma premissa há muito estabelecida pela área de manufatura e acreditar nela é uma tendência visível nos movimentos de qualidade da indústria e setores de serviços (Padrão ISO).

Assim, precisamos investir em melhoria do processo. Precisamos investir em uma melhor gestão da qualidade. Porém, mesmo nos dias de hoje, há muita resistência e

diversas desculpas para não se comprometer com um processo de melhoria. As desculpas mais comuns são:

- *Não tenho tempo para isso*. Um processo eficaz traz ganho de tempo.
- *Não acredito nisso*. Já tentou? Há exemplos de muitos gerentes de projeto que foram muito beneficiados.
- *Isto é muito caro*. Mostre-me seu relatório de custos. Pode custar muito mais caro não investir nisso.
- *Meu cliente está contente, não preciso mudar nada*. Você não gostaria de fazer seu cliente sempre feliz? Não há nada que possa fazer para melhorar?

Muitas pessoas defendem-se afirmando que há diversas metodologias e modelos de desenvolvimento de software diferentes no mercado e que cada um se propõe a resolver um grupo diferente de problemas. Alguns afirmam ainda que metodologias tradicionais tragam burocracia e deixam o processo de desenvolvimento de software ainda mais lento. Outros dizem que metodologias ágeis não conseguem trazer o controle que um gestor de projetos necessita.

Diante deste vasto conjunto de opções, gestores de desenvolvimento de software sentem-se perdidos quando precisam decidir com qual metodologia/modelo sua equipe irá trabalhar. Muitos são guiados pela preferência de seus clientes ou pelos editais. Ou ainda pela tendência do mercado. Mas, com certeza, estas não são as melhores formas de escolher. É preciso conhecer as diferentes metodologias para adotar ou adaptar as melhores práticas para sua equipe e seu tipo de projeto.

1.2. Objetivos

O principal objetivo deste trabalho é estudar duas metodologias ou frameworks de processo de desenvolvimento de software que são consideradas opostas, e até incompatíveis, pela maioria dos gestores de desenvolvimento de software e estabelecer algum nível de relação entre elas. Uma delas, o Capability Maturity Model Integration(CMMI), representa a vertente conservadora e clássica do processo de desenvolvimento. E a outra, a Programação Extrema(XP), representa a nova tendência de metodologia leves ou ágeis.

Durante muito tempo, o desenvolvimento ágil e as melhores práticas do CMMI foram considerados antagônicos entre si. Acredita-se que esta discordância teve origem na diferença extrema entre os primeiros projetos que adotaram ambos os métodos de desenvolvimento. Os primeiros projetos que adotaram as práticas CMMI eram projetos de larga-escala, que lidavam com sistemas de missão crítica e com altos níveis de supervisão gerencial. Já os primeiros projetos que adotaram métodos ágeis de desenvolvimento eram focados em equipes pequenas, com poucos níveis hierárquicos, com requisitos voláteis e focados apenas no desenvolvimento de software[5].

Este trabalho visa mostrar que ambos os métodos não precisam ser considerados opostos e que ambos apresentam bons princípios de desenvolvimento de software. Além disso, este trabalho procura mostrar que muitos princípios apresentados podem ser

considerados compatíveis em alguns casos tentando encontrar compatibilidade entre o modelo clássico CMMI nível de maturidade 2 e a metodologia ágil XP.

1.3. Justificativa

Cada vez mais a comunidade de software vem aceitando, entendendo e aplicando os conceitos ágeis em seus processos de desenvolvimento. Porém, as metodologias ágeis continuam sendo aplicadas em um percentual pequeno de projetos de desenvolvimento de software. Muitos gestores ainda possuem certa desconfiança a projetos que desenvolvem software utilizando metodologias ágeis.

Grandes organizações optam pelo desenvolvimento tradicional, pois traz uma segurança maior para seus clientes, uma vez que estas são as metodologias de mercado. Metodologias tradicionais nos dão idéia de mais controle do processo e menor dependência das pessoas. É mais fácil convencer o cliente da qualidade quando se tem um processo de desenvolvimento eficiente e que produz produtos de qualidade.

Porém, os valores e princípios das metodologias ágeis também podem produzir produtos de qualidade. O desenvolvimento ágil pode ser tão eficaz quanto o tradicional e há muitos casos de sucesso no mercado.

Se a compatibilidade entre diferentes metodologias for academicamente comprovada, as organizações poderão competir com igualdade na busca de contratos de desenvolvimento de software. Clientes e editais não precisarão fixar metodologias de software que deverão ser utilizadas por seus futuros fornecedores de software.

2. Metodologia

A metodologia científica adotada na elaboração deste trabalho é o método hipotético-dedutivo. Este método científico parte de um problema ao qual se apresenta uma solução provisória e passa-se a criticar a solução. É um processo decorrente da identificação de dúvidas e da necessidade de elaborar e construir respostas para esclarecê-las.

Neste trabalho apresentamos o problema causado entre a aparente incompatibilidade entre metodologias de desenvolvimento tradicionais e metodologias ágeis. Porém, acreditamos que há algum nível de compatibilidade e selecionamos dois modelos representativos das duas categorias para compará-los e demonstrar o nível de compatibilidade entre eles.

A teoria-tentativa deste trabalho é que o representante do conjunto de metodologias tradicionais de desenvolvimento, CMMI, é de alguma forma compatível com o representante de metodologias ágeis, a Programação Extrema(XP).

O experimento que este trabalho realiza é a comparação entre as práticas de ambos os métodos de desenvolvimento a fim de mostrar equivalência entre as mesmas.

3. Organização do Trabalho

Este trabalho está organizado da seguinte forma: o capítulo 4 apresenta detalhes das duas metodologias de software que serão analisadas neste trabalho, CMMI e Programação Extrema. No capítulo 5, apresenta a análise de satisfação de XP com as práticas das áreas de processo CMMI nível 2. Por fim, as considerações finais que apresenta a conclusão do trabalho, dificuldades encontradas e a sugestão de trabalhos futuros.

4. Metodologias e modelos de desenvolvimento de software

4.1. CMMI (Capability Maturity Model Integration) Models

CMMI(Capability Maturity Model Integration) Models são coleções de boas práticas e metas de melhorias de processos que as organizações podem utilizar para avaliar e melhorar seus processos[29]. Estas metas e práticas são organizadas em áreas de processo. Cada organização pode escolher quais processos deseja melhorar e focar no que é mais importante.

Atualmente o CMMI é dividido em três modelos:

- **CMMI for Acquisition(CMMI-ACQ)**: Este modelo fornece orientação para organizações que trabalham com cadeia de abastecimento, integrando produtos e consumidores.
- **CMMI for Services(CMMI-SVC)**: Este modelo fornece orientação para organizações que estabelecem, gerenciam e prestam serviços para consumidores e usuários finais.
- **CMMI for Development(CMMI-DEV)**: Este modelo é usado por organizações que desenvolvem produtos e serviços. Fornece orientação para melhorar a eficiência, eficácia e qualidade de seus produtos.

O modelo alvo de nosso estudo é o CMMI for Development.

4.1.1. História CMMI

A força aérea dos Estados Unidos financiou um estudo no SEI para criar um modelo para uso militar com objetivo de avaliar os fornecedores de software. Em 1989, o CMM(Capability Maturity Model) foi publicado como *Managing the Software Process*. CMMI teve sua primeira versão lançada em 2000 como resultado da evolução e integração de outros 3 modelos: *CMM for Software (SW-CMM), Capability Model for System Development (EIA/IS 731)* e *CMM for Integrated Product Development (IPD-CMM).*

A figura abaixo mostra as atividades em torno dos modelos CMM e CMMI ao longo do tempo:

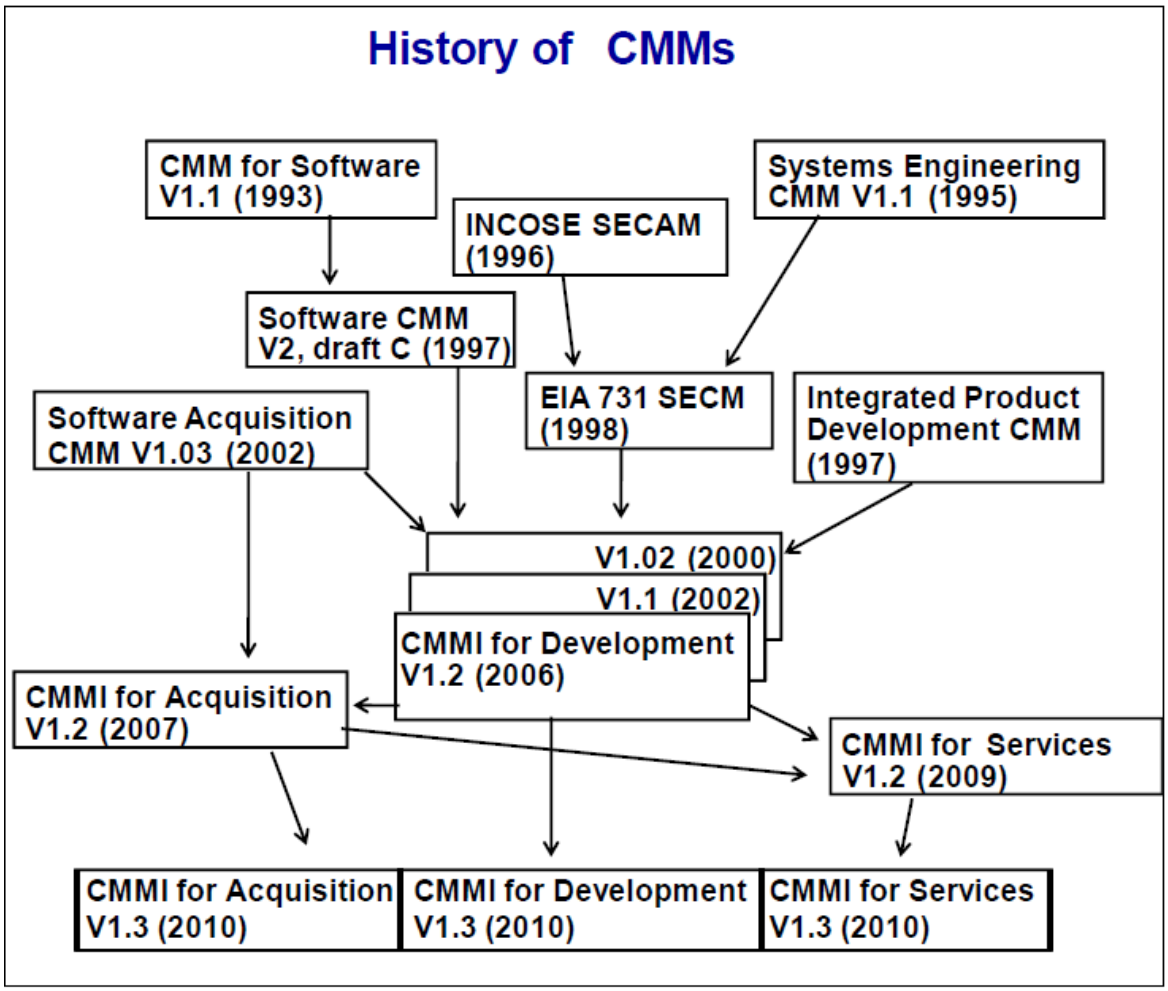

Figura 2 - História dos CMMs retirado de [18]

O projeto CMMI teve como principal objetivo integrar as práticas de melhoria de processos das áreas de software, sistemas de engenharia e produção de produtos para reduzir custos de implantação de modelos multidisciplinares de melhoria de processos.

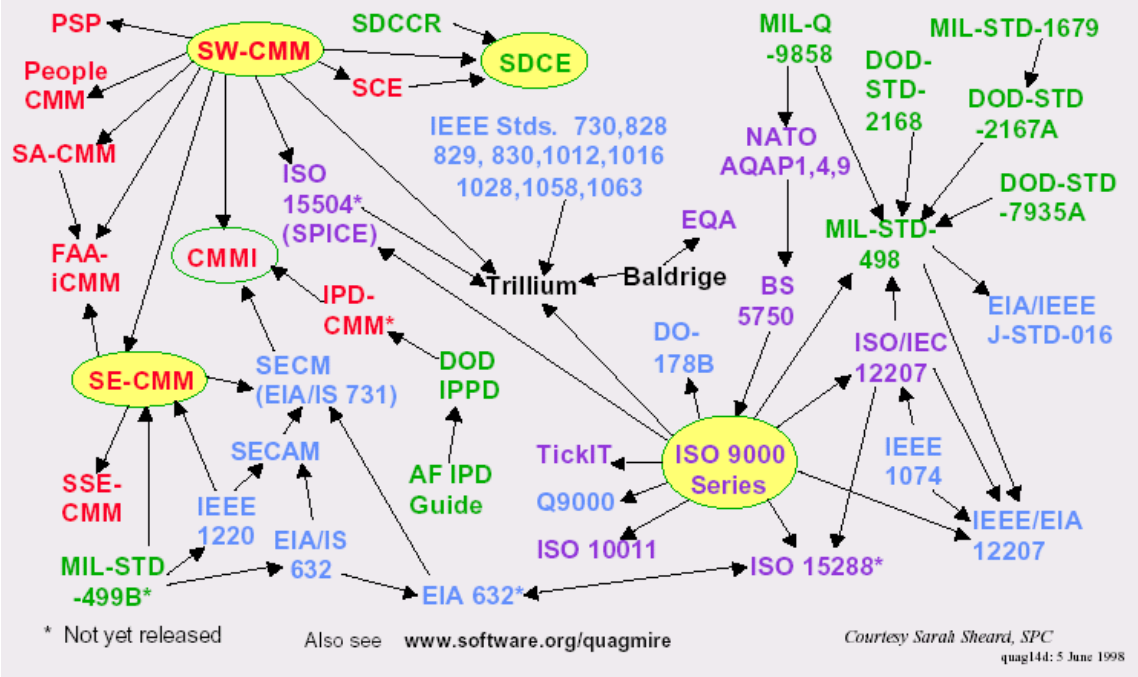

Figura 3 - Frameworks de desenvolvimento de software retirado de [29]

O modelo CMMI possui duas representações: em estágios e contínua.

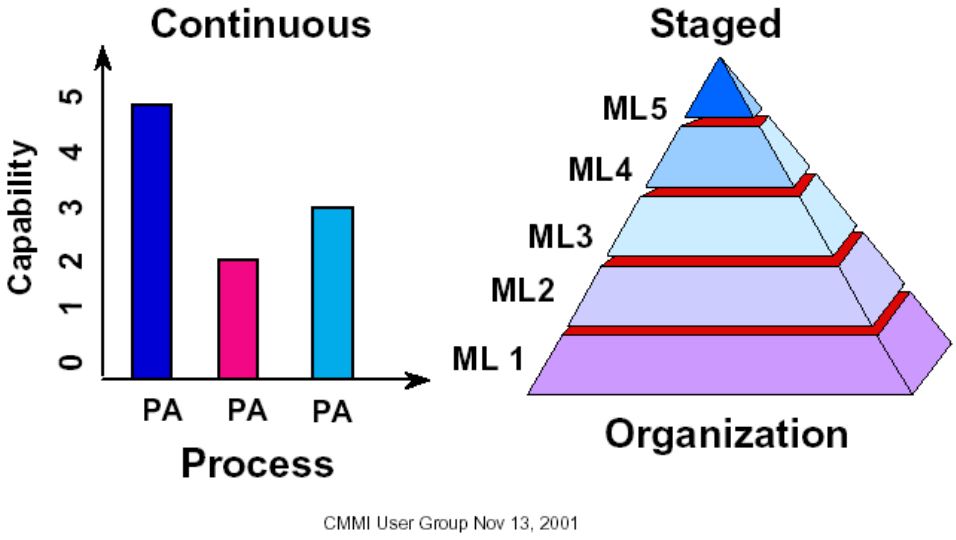

Figura 4 - Duas representações do modelo CMMI retirado de [29]

4.1.2. Modelo em estágios

O modelo em estágios possui um plano de trabalho pré-definido baseado em agrupar e ordenar os processos. O nome "estágios" é devido ao modo como o modelo descreve este plano de trabalho. O plano é descrito como uma série de estágios ou níveis de maturidade (maturity levels). Cada nível de maturidade tem um conjunto de áreas de processos (process areas ou PAs) que indicam onde a organização deve focar seus esforços para melhorar seus processos.

Cada área de processo possui práticas definidas para alcançar as metas estabelecidas. As práticas descrevem a infra-estrutura e as atividades que contribuem para a mais eficiente implementação e institucionalização da área de processo em questão. A organização alcança o nível de maturidade previsto em um estágio quando todas as metas de todas as áreas de processo deste nível foram alcançadas.
Assim, quando ouvimos que uma determinada organização está no nível 3 de maturidade, podemos entender que esta organização foi avaliada e satisfez todas as metas associadas às áreas de processo do nível de maturidade 3.

Os níveis de maturidades previstos para avaliar a evolução das práticas de melhoria na qualidade dos processos das organizações por este modelo são cinco: Inicial, Gerenciado, Definido, Quantitativamente gerenciado e Otimizado.

- Inicial: Não é um nível a ser atingido. Todas as organizações iniciam neste nível. Seus processos são imprevisíveis e ocasionalmente caóticos. O sucesso depende de pessoas altamente competentes e até mesmo heróicas.
- Gerenciado: Nestas organizações, há processos básicos de gerenciamento que controlam prazos, custos e escopos. Porém, o processo é sempre reativo e nunca pró-ativo.
- Definido: Neste nível, encontramos processos de gerenciamento melhores definidos. Há documentação e os procedimentos são padronizados e integrados dentro da organização.
- Quantitativamente gerenciado: Métricas detalhadas dos processos e dos projetos são coletadas. Tanto os processos como os projetos são quantitativamente compreendidos e controlados.
- Otimizado: A melhoria contínua do processo é estabelecida por meio de sua avaliação quantitativa, e da implantação planejada e controlada de tecnologias e idéias inovadoras.

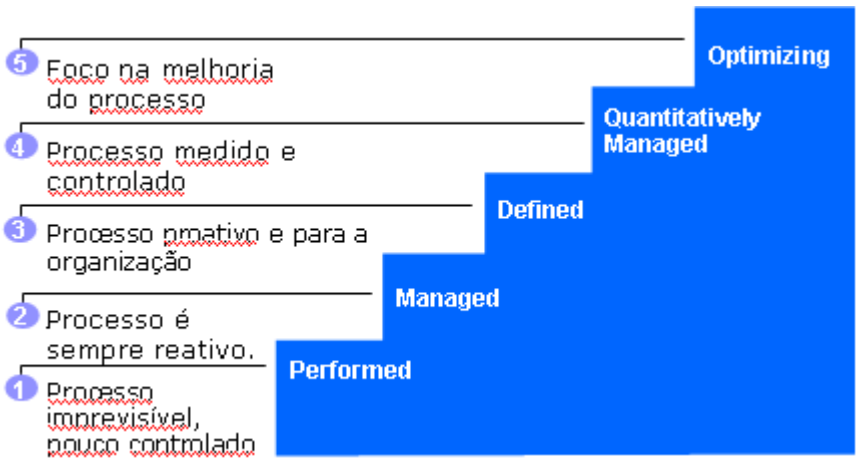

Figura 5 - Níveis de maturidade e seus focos principais

4.1.3. Modelo contínuo

O modelo contínuo é assim chamado, pois não associa estágios à maturidade organizacional. É menos específico ao definir o plano de trabalho e não estabelece ordem na concentração dos esforços para alcançar processos eficientes e eficazes.
Assim como o modelo em estágios, o modelo contínuo também possui áreas de processos e práticas. Porém, as práticas são organizadas de forma diferente. Uma área de processo pode alcançar melhorias individualmente. Isto é, a organização pode escolher focar seus esforços em uma área de processo somente e alcançar suas metas.

A maioria das práticas adotadas por este modelo é genérica, isto é, não são específicas de uma área de processo e podem ser aplicadas a todas as áreas. As práticas genéricas são agrupadas em níveis de capacidade (Capability levels ou CLs). As definições de cada nível de capacidade são intimamente relacionadas às definições dos níveis de maturidade. Uma área de processo é institucionalizada e melhorada implementando as práticas genéricas nesta área. Com o modelo contínuo, cada organização pode montar seu próprio modelo em estágios selecionando as áreas de processo que mais lhe interessam para promover melhorias.

Também é possível que diferentes áreas de processo estejam em diferentes níveis de capacidade. O CMMI inclui 6 níveis de capacidade que são numerados de 0 a 5. Os níveis de capacidade indicam como a organização está avaliada em uma área de processo específica.

- Incompleto(CL 0): indica que uma ou mais metas desta área de processo não são satisfeitos pela organização.
- Implementado(CL 1): indica que as práticas base desta área de processo são realizadas pela organização.
- Gerenciado(CL 2): indica que a área de processo teve suas práticas institucionalizadas. Os projetos seguem um plano de trabalho documentado.
- Definido(CL 3): indica que as práticas em uma determinada área de processo foram padronizadas e institucionalizadas na organização.
- Quantitativamente gerenciado(CL 4): indica que uma área de processo é quantitativamente gerenciada, isto é, utiliza-se estatística e métodos quantitativos para controlar subprocessos.
- Otimizado(CL 5): indica que os métodos quantitativos estabelecidos no CL 4 são utilizados para otimizar e inovar o processo buscando melhorias contínuas.

4.1.4. Compatibilidade entre modelo contínuo e em estágios

Por que duas representações?

Os modelos de software que foram base para a construção do modelo CMMI possuem representações distintas. O CMM para software possui uma representação em estágios.

O EIA/IS 731 possui representação contínua. Já o IPD tem representação híbrida, ou seja, combina aspectos interessantes das duas representações.

Assim, o time CMMI achou que adotar uma das duas representações era uma decisão difícil e conseguiu que o CMMI fosse representado nestas duas formas, modelo em estágios e modelo contínuo. Ambos os modelos possuem o mesmo objetivo comum, melhoria na qualidade dos processos das organizações, porém possuem enfoques diferentes.

Compatibilidade entre modelo contínuo e em estágios

É importante ressaltar que apesar das duas representações, o CMMI é um modelo único. Logo, as duas representações são equivalentes e é possível sair de uma representação para outra sem adaptações. Por exemplo: uma organização, que adotou o modelo contínuo, alcançou o nível de capacidade 2 (CL 2) nas sete áreas de processo que formam o nível de maturidade 2(no modelo em estágios), então podemos dizer que esta organização está no nível 2 de maturidade. Veja a figura abaixo:

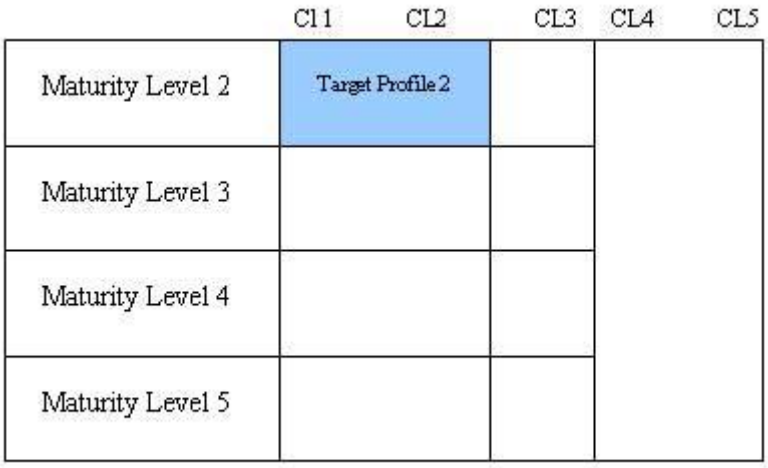

Figura 6 - Compatibilidade entre modelo contínuo e por estágios adaptado de [29]

Da mesma forma, se uma organização alcançou o nível de capacidade 3(CL 3) nas setes áreas de processo que formam o nível de maturidade 2 e nas 14 áreas que formam o nível de maturidade 3(um total de 21), então podemos afirmar que esta organização está no nível de maturidade 3. Veja a figura abaixo:

Figura 7 - Compatibilidade entre modelo contínuo e por estágios adaptado de [29]

Já para os níveis de maturidade 4 e 5, estabelecer equivalência entre as duas representações não é tão intuitiva. Por exemplo, o nível de maturidade 4 não requer que as atividades previstas estejam implementadas em todas as áreas de processo mapeadas pelos níveis 2 e 3. Uma organização que está no nível de maturidade 4 pode gerenciar quantitativamente um conjunto de áreas de processo e outra organização pode gerenciar quantitativamente outro grupo de áreas de processo. Estes conjuntos de áreas podem totalmente distintos entre si e mesmo assim as duas organizações encontram-se no nível de maturidade 4.

O foco do nosso estudo será na versão em estágio do CMMI-DEV por ser a versão mais utilizada pelas organizações.

4.1.5. Componentes do Modelo CMMI

Os componentes das áreas de processo podem ser classificados em 3 categorias: obrigatórios esperado ou informativo.

Os componentes obrigatórios são aqueles que são essenciais para o alcance das melhorias propostas em uma determinada área de processo. São mais conhecidos como metas específicas e genéricas (*Specific Goals*(SG) e *Generic Goals*(GG)).

Já os componentes esperados são aqueles que descrevem quais são as atividades importantes a serem realizadas com o objetivo de obter um determinado componente. São mais conhecidos como práticas genéricas e específicas (*Generic Pratices*(GP) e *Specific Pratices*(SP)). Antes que uma meta seja considerada satisfeita, suas práticas devem estar presentes nos processos da organização.

E os componentes informativos são aqueles que têm como objetivo ajudar na compreensão dos componentes obrigatórios e esperados do CMMI. O material informativo é essencial para o entendimento do modelo e podem ser simples caixas de textos, explicações detalhadas, subpráticas, notas e exemplos de trabalho.

A figura abaixo mostra como se relacionam os componentes que compõem o modelo CMMI.

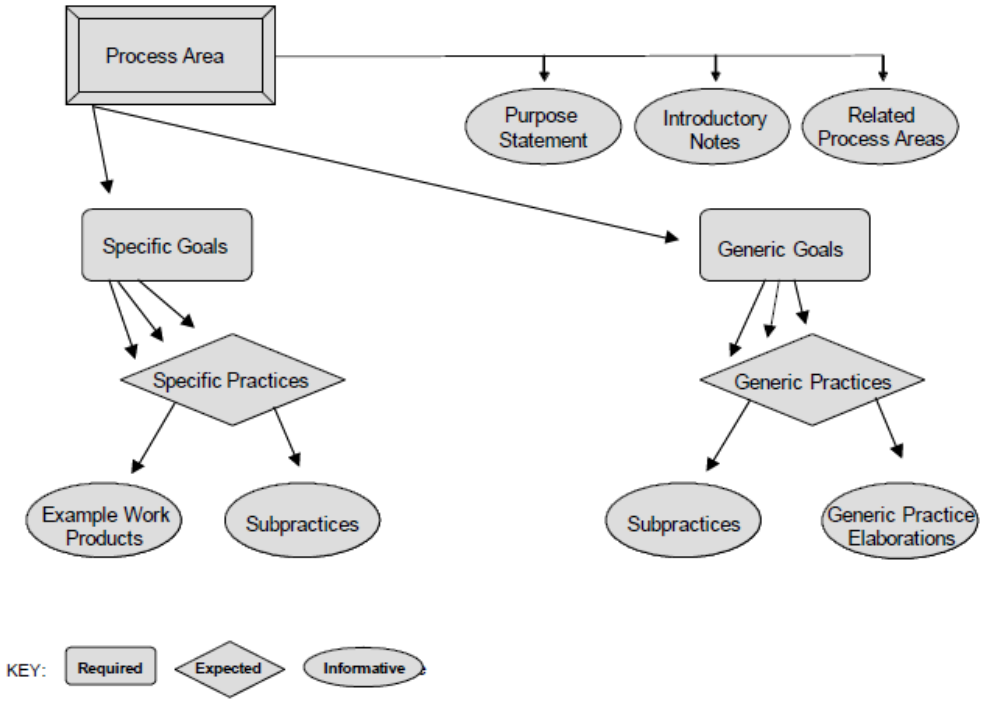

Figura 8 - Relação entre os componentes CMMI retirado de [18]

4.1.5.1. Áreas de processo(Process Area) ou PAs

Áreas de processo são aglomerados de práticas relacionadas a uma área da organização[29]. Quando estas práticas são realizadas coletivamente, satisfazem um conjunto de metas(goals) consideradas importantes para a realização de melhorias significativas na área.
O modelo CMMI for Development conta com 22 áreas de processo que estão em ambas as formas de representação. No modelo em estágio, as áreas de processo são agrupadas por categoria:

- Gerência de Processo
- Gerência de Projeto
- Engenharia
- Suporte.

As áreas de processo de 3 categorias(com exceção de Suporte) são relacionadas entre si. Já as áreas de processo da categoria Suporte são independentes uma das outras e de todas as outras áreas de processo.
A tabela abaixo mostra quais são as áreas de processo de cada categoria:

Categoria	Áreas de Processo
Gerência de Processos(Process Management)	Foco no processo organizacional(Organizational Process Focus - OPF)
	Definição de processo organizacional(Organizational Process Definition - OPD)
	Treinamento organizacional (Organizational Training - OT)
	Performance do processo organizacional(Organizational process performance - OPP)
	Inovação e melhoria organizacional(Organizational Innovation and Deployment - OID)
Gerência de projetos(Project Management)	Planejamento de projeto(Project Planning - PP)
	Controle e monitoramento de projetos(Project Monitoring and Control - PMC)
	Acordo de gestão de fornecedores (Supplier Agreement Management - SAM)
	Gerência integrada de processo(Integrated Project Management - IPM)
	Gerência de riscos (Risk Management - RSKM)
	Gerência quantitativa de projetos (Quantitative Project Management - QPM)
Engenharia(Engineering)	Gerência de Requisitos(Requirements Management - REQM)
	Desenvolvimento de requisitos(Requeriments Development - RD)
	Solução técnica(Tecnical Solution - TS)
	Integração de produto(Product Integration - PI)
	Verificação (Verification - VER)
	Validação(Validação - VAR)
Suporte(Support)	Métricas e análise(Measurement and Analysis - MA)
	Garantia de qualidade de produtos e processos(Process and Product Quality Assurance - PPQA)
	Gerência de Configuração(Configuration Management - CM)
	Análise de decisão e resolução(Decision Analysis and Resolution - DAR)
	Análise e resolução causal (Causal analysis and resolution - CAR)

A representação em estágios do CMMI apresenta a seguinte distribuição das áreas de processo por nível de maturidade:

Nível de Maturidade	Foco	Áreas de Processo
5 - Otimizado	Melhoria contínua do processo	Inovação e melhoria organizacional Análise e resolução causal

4 - Quantitativamente gerenciado	Gestão quantitativa	Performance do processo organizacional Gerência quantitativa de projetos
3 – Definido	Padronização do processo	Desenvolvimento de requisitos Solução técnica Integração de produto Verificação Validação Foco no processo organizacional Definição de processo organizacional Treinamento organizacional Gerência integrada de processo Gerência de riscos Análise de decisão e resolução
2- Gerenciado	Gerenciamento básico	Gerência de Requisitos Planejamento de projeto Controle e monitoramento de projetos Acordo de gestão de fornecedores Métricas e análise Garantia de qualidade de produtos e processos Gerência de Configuração
1 – Inicial		

4.1.5.2. Definições de finalidade (Purpose Statements)

A definição de finalidade é um componente informativo e descreve o propósito de determinada área de processo. Por exemplo, podemos citar a definição de finalidade da área de processo Definição de processo organizacional. A finalidade da área de Definição de processo organizacional é "estabelecer e manter um conjunto ativo de processos da organização, ambiente de trabalho padrão e regras e diretrizes para equipe"[18].

4.1.5.3. Notas introdutórias(Introductory Notes)

A seção de notas introdutórias de uma área é um componente informativo e descreve os principais conceitos abordados nesta área. Podemos citar como exemplo a uma nota da seção introdutória da área de Controle e monitoramento de projetos: "Quando o estado real se desvia significativamente dos valores esperados, ações corretivas são tomadas as medidas adequadas"[18].

4.1.5.4. Áreas de processo relacionadas(Related Process Area)

A seção áreas de processo relacionadas é um componente informativo e lista as outras áreas de processo que estão relacionadas de alguma maneira a uma determinada área. Também estabelece o nível de relacionamento entre as áreas. Se analisarmos a seção áreas de processo relacionadas da área Planejamento de Projeto vamos encontrar uma referência à área Gerência de riscos.

4.1.5.5. Metas Específicas(Specific Goals)

Uma meta específica descreve uma característica única que deve estar presente em seu processo para satisfazer os objetivos da área de processo. Para exemplificar uma meta específica, vamos falar da área de processo de Validação(Validation). O objetivo desta área de processo é demonstrar que o produto ou os componentes do produto cumpre sua utilização prevista, quando colocado no ambiente desejado.

Esta área possui duas metas específicas (SG):

SG1 - Preparar para validação.
SG2 - Validar produto.

Esta descrição da característica única é um componente obrigatório. Já suas notas associadas são componentes informativos.

4.1.5.6. Metas Genéricas(Generic Goals)

Metas genéricas são chamadas assim, pois podem ser aplicadas a várias áreas de processo. Descrevem objetivos genéricos que devem estar presentes no processo de melhoria para institucionalizar os processos que implementam a área.

É um componente obrigatório e é utilizado, durante a avaliação, para determinar se a organização implementa ou não processos que satisfazem esta área.

Vamos citar um exemplo de meta genérica para o nível de maturidade 2(Gerenciado):

GG2: O processo é institucionalizado como um processo gerenciado.

4.1.5.7. Práticas Específicas(Specific Pratices)

Uma prática específica descreve uma atividade que é considerada importante para o alcance de sua meta específica associada.

Para exemplificar a prática específica, vamos voltar a falar da área de processo de Validação(Validation) que utilizamos no exemplo de meta específica. Como já citamos anteriormente, esta área possui duas metas específicas:

SG1 - Preparar para validação.
SG2 - Validar produto.

Para cada uma das metas específicas para a área de processo de validação, temos suas práticas específicas(SP):

SG1 - Preparar para validação.
SP1.1 - Selecionar os produtos para validação(Por exemplo: Sistema, unidades de hardware, software).
SP1.2 - Estabelecer o ambiente de validação.
SP1.3 - Estabelecer os critérios e procedimentos para validação(Performance, critérios de aceitação do usuário, requerimentos do produto).

SG2 - Validar produto
SP2.1 - Executar validação
SP2.2 - Analisar os resultados da validação

Esta descrição da característica única da atividade é um componente esperado. Já suas notas associadas são componentes informativos.

4.1.5.8. Práticas Genéricas(Generic Pratices)

Práticas genéricas são chamadas assim, pois podem ser aplicadas a várias áreas de processo. As práticas genéricas estão associadas a metas genéricas. Descrevem atividades genéricas que são consideradas importantes para o alcance de sua meta genérica associada.

Para exemplificar a prática genérica, vamos voltar ao exemplo de meta genérica que citamos anteriormente:

GG2: O processo é institucionalizado como um processo gerenciado.

Para atingir esta meta, temos 10 práticas genéricas que nos levam ao nosso objetivo:

GP2.1 - Estabelecer uma Política Organizacional (Estabelecer e manter uma política organizacional para planejar e executar o processo);
GP2.2 – Plano do processo(Estabelecer e manter o plano para executar o processo);
GP2.3 – Fornecer recursos(Fornecer recursos adequados para executar o processo e desenvolver os produtos de trabalho).
GP2.4 – Atribuir responsabilidade(Atribuir responsabilidade e autoridade pela execução do processo e desenvolvimento dos produtos de trabalho).
GP2.5 – Treinar pessoas(Treinar as pessoas para que possam desempenhar seus papéis na execução e suporte do processo).

GP2.6 – Gerenciar configurações(O produto de trabalho deve estar sob nível apropriado de gerência de configuração).
GP2.7 – Identificar e envolver as partes interessadas(Identificar e envolver as partes interessadas no plano do processo).
GP2.8 - Acompanhar e controlar o processo(Monitorar e controlar a execução do processo e comparar os resultados com o que foi planejado. Tomar ações corretivas quando há desvio).
GP2.9 – Avaliar aderência(Avaliar a aderência do processo com sua descrição, padrões e processos e identificar os descumprimentos).
GP2.10 – Revisar o estado do processo com o mais alto nível da gestão(Revisar as atividades, objetivos e resultados dos processos com o responsável pela gestão).

Esta descrição da característica única da atividade é um componente esperado. Já suas notas associadas são componentes informativos.

4.1.5.9. Exemplos de produtos de trabalho(Example Work Products)

A seção de exemplo de produtos de trabalho apresenta amostras de saídas de uma determinada prática específica. Um exemplo de produto de trabalho da prática específica *Monitorar parâmetros de planejamento de projeto* da área Controle e monitoramento de projetos é o documento de *Registros de desvios significativos*.

4.1.5.10. Subpráticas(Subpratices)

A subprática é descrição detalhada que provê um guia para interpretar e implementar uma prática genérica ou específica. É um componente informativo que tem como objetivo apenas prover idéias que podem ser úteis durante o processo de melhorias.

4.1.5.11. Elaborações de práticas genéricas(Generic Pratice Elaborations)

Elaborações de práticas genéricas aparecem depois das práticas genéricas para fornecer orientação de como as práticas genéricas podem ser aplicadas em uma determinada área de processo. É um componente informativo do modelo.

Por exemplo, uma elaboração de prática genérica para a prática *Estabelecer e manter uma política organizacional para planejamento e executar o processo* da área de processo *Planejamento do Projeto* é: "Esta política estabelece expectativas organizacionais para estimar parâmetros de planejamento, estabelecer compromissos internos e externos, e desenvolvimento do plano para gerenciar o projeto"[18].

4.2. Metodologias Ágeis

Na primeira década do século 21, os métodos ágeis foram se tornando cada vez mais populares e ganhando espaço em diversos segmentos da indústria de software. As metodologias ágeis trouxeram uma nova perspectiva para o desenvolvimento de software tirando o foco do processo e colocando no produto. Isso exigiu mudanças significativas em características importantes dos métodos de desenvolvimento de software tradicionais, por isso os métodos ágeis tornaram-se tão polêmicos.

Em fevereiro de 2001, um grupo de 17 líderes de desenvolvimento, que adotavam práticas próprias e estavam no contra-fluxo dos padrões adotados pelo mercado, se reuniram para discutir suas formas de trabalho. Isoladamente, eles já tinham percebido que seus métodos mostravam-se mais eficientes do que nos métodos tradicionais. Juntos, esperavam construir uma nova metodologia de desenvolvimento que agregasse as melhores práticas de cada um.

O resultado desta reunião não foi exatamente o que o grupo esperava. Uma nova metodologia não foi elaborada e conclui-se que desenvolver software é uma atividade muito complexa para ser padronizado. Mas identificar alguns princípios que seriam determinantes para um projeto de desenvolvimento de software de sucesso. Os princípios identificados deram origem para o Manifesto Ágil[24].

O Manifesto Ágil é a base para os métodos ágeis, apresentando quatro valores principais que um método ágil deve ter:

- **Indivíduos e iterações** são mais importantes que processos e ferramentas
- **Software funcionando** é mais importante do que documentação abrangente
- **Colaboração com o cliente** é mais importante do que negociação de contratos
- **Responder a mudanças** é mais importante do que seguir um plano

Estes valores ressaltam o que tem mais valor para métodos ágeis: foco no produto final e nos interesses de negócio do cliente[24].

O Manifesto Ágil apresenta uma nova abordagem para o desenvolvimento de software. Com base nos valores e princípios do manifesto, surgiram abordagens mais específicas e com idéias diferentes. É muito comum ver a troca de conhecimento e a incorporação de práticas entre as diferentes abordagens. Neste trabalho, vamos estudar os princípios e valores específicos da abordagem proposta por Programação Extrema (XP).

4.2.1. Programação Extrema(XP)

Programação Extrema (XP) é uma das abordagens de desenvolvimento ágil que mais recebeu atenção na última década. Seus valores e práticas são resultado do trabalho de Kent Beck, Martin Fowler e Ron Jeffries em um projeto crítico na Chrysler. Com sua larga experiência em desenvolvimento de software com Smalltalk e seu conhecimento nos valores ágeis, Beck selecionou um conjunto de práticas que já haviam se mostrado eficientes em outros de seus projetos e as aplicou em conjunto.

Em 1999, Kent publicou *Extreme Programming Explained: Embrace Change* que apresentou e oficializou a metodologia que se tornaria tão famosa e polêmica[25]. Nesta primeira versão, XP ainda se apresentava pouco flexível e abrangente. Apenas equipes pequenas conseguiriam utilizar seus princípios e práticas. Em 2004, a segunda edição foi publicada por Kent em conjunto com sua esposa. As modificações incluídas tornaram o método mais abrangente e flexível, tornando possível sua adoção por equipes maiores.

Segundo Kent Beck[25], a programação extrema é formada por valores, princípios e práticas. As práticas são técnicas utilizadas no dia-a-dia. Já os valores são critérios mais amplos utilizados para julgar uma determinada situação. "Os valores dão razão às práticas, enquanto as práticas evidenciam os valores. Para preencher o espaço vazio entre valores e práticas, existem os princípios"[21]. Os princípios estabelecem a ligação entre valores e práticas. Também servem de guia para situações em que as práticas propostas não se aplicam.

4.2.1.1. Valores

Comunicação

A comunicação já foi apresentada como fator principal no Manifesto Ágil através do valor "**Indivíduos e iterações** são mais importantes que processos e ferramentas"[24]. XP acredita que uma comunicação eficiente poderia evitar muitos problemas em um projeto de desenvolvimento de software. Por isso, XP favorece a comunicação entre toda a equipe através de atividades colaborativas. Quando estivermos estudando as práticas propostas por XP, vamos identificar muitas que proporcionam este ambiente de trabalho colaborativo como **Time Completo**, **Área de Trabalho Informativa** e **Sentar Junto**.

Simplicidade

Segundo Kent Beck, a Simplicidade é o valor intelectual mais intenso de XP[25]. Os membros da equipe devem sempre procurar pela solução mais simples. A simplicidade impacta diretamente no valor "**Responder a mudanças** é mais importante do que seguir um plano" do Manifesto Ágil[24]. As mudanças são mais facilmente absorvidas com abordagens simples, pois gasta-se menos para produzir uma solução simples e depois sofisticá-la de acordo com as mudanças.

Coragem

A coragem, como valor de XP, é importante para inovar, manter a simplicidade nas soluções e a humildade ao assumir que o processo de desenvolvimento de software é dinâmico e muitas decisões precisam ser revistas ao longo do tempo. Assim, é preciso investir em **Refatoração** do código e propor **Contratos de Escopo Variável**. Os princípios de *humanidade* e *aceitação de responsabilidade* valorizam as pessoas e aumentam suas autoconfianças e coragem.

Feedback

O feedback constante pode ser considerado como um componente principal do sucesso de XP em relação à rápida adaptação a mudanças. XP trabalha com ciclos curtos e constantes de feedback em todos os aspectos do desenvolvimento. Todas as partes interessadas no projeto são encorajadas a identificar impedimentos e trabalhar em sua remoção. As práticas **Programação Pareada** e **Desenvolvimento Dirigido por Testes** evidenciam a importância do feedback.

Respeito

O respeito é valor principal no qual os demais se apóiam. Não é possível existir uma comunicação eficiente e aberta, troca constante de feedback verdadeiro e um time corajoso que busca sempre as soluções mais simples sem respeito. Todos devem manter respeito entre si em relação aos produtos de trabalho.
A prática **Programação Pareada** é um exercício diário de respeito. Os princípios da *humanidade, diversidade* e *aceitação* da responsabilidade evidenciam o valor do respeito entre todos os envolvidos no projeto.

4.2.1.2. Princípios

Humanidade

O desenvolvimento de software é feito por pessoas. É preciso levar em consideração as necessidades individuais de cada um e procurar balancear com as necessidades do projeto e da equipe como um todo.

Economia

Todos os envolvidos no projeto devem se preocupar com os aspectos econômicos do projeto. Para XP, é preciso agregar valor ao negócio do cliente em menor tempo possível. Por isso, é o cliente que deve atribuir prioridade às **Histórias** nas reuniões de planejamento de release.

Benefício Mútuo

As atividades executadas, dentro de um projeto XP, devem trazer benefício a todos os envolvidos.

Auto-semelhança

O princípio da auto-semelhança sugere que a estrutura ou conceito macro de uma solução possa ser aplicado em outro contexto.

Melhoria

O princípio da melhoria valoriza atividades que são refinadas ao longo do tempo. É melhor produzir uma boa solução e colocá-la rapidamente em produção e refiná-la do que buscar a perfeição.

Diversidade

A equipe do projeto deve ser composta por diferentes habilidades, opiniões e perspectivas. Em um cenário assim, o conjunto de idéias e soluções é mais amplo.

Reflexão

A equipe deve refletir constantemente sobre o trabalho realizado. Através da reflexão é possível identificar os pontos de falha, de sucesso e buscar soluções para melhorar.

Fluxo

O processo de desenvolvimento de software deve trabalhar com um fluxo contínuo de entregas. Quanto maior o tempo gasto trabalhando, mais tempo se levará para descobrir se este foi realizado com sucesso ou teve falhas. Quanto maior a falha, mais difícil e caro será corrigi-la.

Oportunidade

A equipe deve encarar mudanças como oportunidades para aprender e melhorar.

Redundância

O princípio da redundância tem como objetivo reduzir chances de erros.

Falha

Toda falha deve ser encarada como uma oportunidade de aprendizado. A equipe deve ter coragem para tentar soluções diferentes mesmo que algumas resultem em falha.

Qualidade

Um projeto de XP não considera a qualidade como uma variável de controle. Sacrificar a qualidade não é uma opção. A equipe deve buscar sempre o aumento da qualidade, pois trará mais confiança e satisfação para o cliente, maior velocidade para novas implementações e motivação para a equipe.

Passos Pequenos

Este princípio complementa o princípio Fluxo. A equipe deve se perguntar sempre "Qual o mínimo que devo fazer para garantir que estou na direção certa?" Desta forma, minimizamos os riscos e os possíveis erros tornando o custo para corrigi-los bem menor.

Aceitação da Responsabilidade

Cada membro da equipe deve aceitar suas responsabilidades. Quando há participação na atribuição das responsabilidades e estimativas, temos um time mais comprometido e motivado.

4.2.1.3. Práticas

As práticas de XP foram reformuladas por Kent Beck em sua segunda versão. Foram classificadas em dois grupos. O primeiro grupo é formado pelas práticas primárias que podem ser implementadas individualmente promovendo a introdução da equipe à metodologia XP. O segundo grupo é formado pelas práticas corolárias que são mais difíceis de serem aplicadas. Para o sucesso das práticas corolárias, a equipe deve ser um pouco mais experiente com a metodologia.

Práticas Primárias

Sentar Junto

A equipe deve possuir um local de trabalho amplo onde todos possam ficar juntos. A proximidade é muito importante para uma comunicação eficiente. O espaço deve permitir que todos sentem juntos para discutir soluções e praticar reflexões.

Time Completo

A equipe deve possuir todas as habilidades analíticas e técnicas necessárias para sucesso do projeto.

Área de Trabalho Informativa

A área de trabalho, além de ampla e permitir que todos sentem junto, deve ser informativa. Deve refletir o progresso do projeto através de gráficos, tabelas, cartões de histórias e tudo mais que a equipe achar necessário e/ou importante.

Trabalho Energizado

Os membros da equipe devem trabalhar somente enquanto forem produtivos e se manter energizados. Fazer horas-extra deve ser exceção e não a regra. Deve-se respeitar o equilíbrio entre a vida pessoal e profissional dos membros da equipe.

Programação Pareada

Esta prática está intimamente ligada ao princípio da redundância. Os desenvolvedores devem trabalhar em pares para realizar tarefas. As duplas devem ser trocadas várias vezes durante o projeto e, até, várias vezes ao dia. Isso promove o compartilhamento de idéias entre os membros da equipe.

Histórias

As funcionalidades desejadas para o software são descritas em cartões de papel que podem ser manuseados facilmente por qualquer membro da equipe ou do cliente.

O ideal é que o próprio cliente escreva suas *histórias* nestes cartões de papel e atribua prioridades a elas. Assim, será mais fácil atender às expectativas do cliente e agregar valor ao negócio no menor tempo possível.

Na reunião de planejamento, ou Jogo do Planejamento como XP denomina, as histórias e suas prioridades são analisadas pelo time e os desenvolvedores devem atribuir estimativas às histórias.

Ciclo Semanal

Esta prática sugere que os trabalhos da equipe XP sejam planejados a cada semana. A cada semana, o time se reúne para refletir sobre o trabalho realizado na semana anterior, analisar o realizado versus o planejado do projeto, sugerir pontos de melhoria e planejar e priorizar as histórias com o cliente.

Ciclo Trimestral

Os releases do software são planejados de forma trimestral. Um release está relacionado a um tema. O tema é diferente de história pois é mais abrangente.

Folga

Durante o Jogo do Planejamento, deve-se ser considerado que as estimativas não são totalmente precisas. É preciso introduzir folgas no formato de tarefas menores ou menos importantes entre as tarefas principais para assegurar que estas sejam entregues. Caso haja atrasos, estas tarefas menores podem ser retiradas e o compromisso com o release mantido.

Build Ágil

O desenvolvimento de software em XP deve ser feito de forma incremental. Ao adicionar uma funcionalidade ao sistema, seus desenvolvedores devem garantir que toda a bateria de testes continua rodando e a qualidade do sistema foi mantida. O build do

sistema e a execução de todos os testes devem ser realizados de forma rápida, em até 10 minutos.

Integração contínua

O código-fonte do sistema deve ser armazenado em repositório único, onde todos tenham acesso. Cada dupla de desenvolvimento deve integrar o seu código à este repositório após o término de cada tarefa. Ao adicionar uma funcionalidade ao sistema, devem garantir que toda a bateria de testes continua rodando e a qualidade do sistema foi mantida. Assim, a dificuldade de integração do código e seus impactos diminuem consideravelmente.

Desenvolvimento Dirigido por Testes

XP tem os testes automatizados como um grande aliado para a manutenção da qualidade do software produzido pelo seu time. Os testes automatizados garantem que, a cada nova integração, o sistema estará sendo verificado integralmente e qualquer quebra de funcionalidade será identificada em minutos.

Design Incremental

Esta prática está ligada ao princípio de passos pequenos. Deve-se evitar a complexibilidade e generalizações desnecessárias visando possíveis mudanças de requisitos no futuro.

Práticas Corolárias

Envolvimento Real com o Cliente

O cliente deve fazer parte do time. Ele deve conhecer as necessidades dos reais usuários do sistema para escrever histórias e definir prioridades.

Implantação Incremental

Grandes implantações possuem um nível de risco muito elevado. É mais seguro agregar novas funcionalidades ou substituir sistemas legados de forma incremental.

Continuidade da Equipe

Durante o processo de desenvolvimento de software, há a tendência de tratar as pessoas como recursos substituíveis. No entanto, não devemos ignorar o valor das interações e relações entre as pessoas do time. Mantenha as pessoas eficientes trabalhando juntas.

Diminuição da Equipe

Conforme a equipe melhora sua capacidade de produção, é natural que reduza a carga sobre os membros. Neste cenário, pode-se liberar um dos recursos para trabalhar em outro projeto enquanto os demais continuarão trabalhando normalmente.

Análise de Causa Inicial

A análise de causa inicial exige que quando um defeito for encontrado, este seja corrigido assim como suas causas.

Código Compartilhado

O repositório de código-fonte é compartilhado por toda a equipe e qualquer um pode inserir mudanças, melhorias ou novas funcionalidades em qualquer parte do código. Não há responsáveis por partes do sistema. A equipe é responsável pelo sistema inteiro.

Código e Testes

Em XP, a equipe foca-se em desenvolver código e testes automatizados. A documentação deve ser evitada ao máximo, pois os testes automatizados representam uma documentação atualizada do que o código representa.

Repositório de código unificado

O repositório de código deve ser único para toda a equipe. Quanto maior o número de versões do mesmo código, maior será o trabalho de sincronização e mais difícil será o entendimento da equipe. A manutenção de um repositório de código único está intimamente associada às práticas primárias Integração contínua e Build Ágil.

Implantação Diária

A implantação diária aconselha que novas versões do sistema sejam colocadas em produção toda noite. Este procedimento aumenta a velocidade do Feedback entre o que o usuário espera e o que o programador entrega. Porém, isso só será possível se a equipe estiver muito bem alinhada com as práticas Build Ágil, Integração contínua e Desenvolvimento Dirigido por Testes.

Contrato de Escopo Negociável

Em XP, o escopo deve ser negociável para que a equipe possa trabalhar naquilo que é mais importante para o cliente.

Pague pelo uso

Esta prática sugere que o cliente pague cada vez que o sistema é utilizado. No modelo tradicional, o cliente paga a cada release. Se este método for adotado por XP, teremos um conflito de interesses entre equipe e cliente. A equipe deseja fazer cada vez mais releases, seguindo as práticas de pequenos passos e design incremental. E o cliente irá querer minimizar o número de releases e maximizar o número de funcionalidades a cada release.

4.2.1.4. Papéis e Fase em XP

A prática Time Completo sugere que a equipe XP deve possuir todas as habilidades analíticas e técnicas necessárias para sucesso do projeto. Kent Beck descreve papéis importantes dentro de uma equipe, mas nem todos precisam existir ou serem atribuídos a pessoas diferentes[25].

Os papéis previstos são **programadores** responsáveis por estimar, implementar e testar; **analistas de negócio** que auxiliam o cliente a escrever histórias; **analistas de testes** que identificam cenários de teste; **arquitetos** que procuram e executam refatorações de larga escala; **projetistas de interação** que avaliam a utilizam do sistema pelos usuários; **gerente de projeto** que atuam como facilitadores do trabalho da equipe; **gerente de produto** que escrevem e priorizam histórias e definem os objetivos dos ciclos de projeto; **redatores técnicos** que escrevem a documentação para o usuário final; **executivos** que preocupam-se com questões de alto nível; **recursos humanos** que resolvem problemas burocráticos; e, **usuários** que fornecem feedback para o ajuste do sistema e novas histórias.

Entre os programadores, há dois papéis de destaque Tracker e Coach. O **coach** é, usualmente, o programador mais experiente da equipe. Ele é responsável por garantir que o time está realizando as práticas propostas. Já o **tracker** é o responsável por fornecer informações de progresso do projeto. Ele seleciona as métricas que julgar mais adequadas ao seu projeto e exibe as informações através de gráficos, tabelas e pôsteres tornando a área de trabalho informativa.

5. Análise de compatibilidade CMMI 2 x Programação Extrema

5.1. A análise

No capítulo 4, descrevemos o modelo do CMMI e seus componentes. Sabemos que uma organização alcança o nível de maturidade previsto em um estágio quando todas as metas de todas as áreas de processo deste nível foram alcançadas. Cada meta está associada a práticas que, se implementadas, ajudam a alcançá-las. As práticas não possuem implementação obrigatória pelo CMMI, mas a organização deve apresentar alternativas de implementação para satisfazer determinada meta.

Diante da importância das práticas para o alcance das metas propostas pelo CMMI, será realizada uma análise de satisfação das práticas de XP contra as práticas do CMMI nível 2. Tomaremos como verdadeiro que se as práticas do nível 2 forem satisfeitas, as metas também serão.

5.1.1. Escopo da análise

O escopo da análise será apenas o nível 2 do CMMI, que abrange as áreas de processo: Medição e Análise (MA), Gerência da configuração (CM), Monitoramento e Controle de Projeto (PMC), Planejamento de Projeto (PP), Garantia da Qualidade de Processo e Produto (PPQA), Gestão de requisitos (REQM) e Gestão de Contrato com Fornecedores(SAM). Se as práticas destas áreas de processo forem satisfeitas, XP atinge as metas propostas pelo nível 2.

Como a metodologia XP é focada no desenvolvimento de software, a área de processo Gestão de Contrato com Fornecedores(SAM) não será considerada nesta análise pois tem como objetivo fornecer subsídios para gerenciar a aquisição de produtos de fornecedores.

5.1.2. Metodologia utilizada

A análise foi realizada sobre cada prática das áreas de processo que compõem o nível 2 de maturidade CMMI e uma classificação foi atribuída a cada uma delas indicando o nível de satisfação por XP. Os níveis de satisfação foram:

- Satisfaz: se a prática foi totalmente satisfeita por XP ou XP apresentou uma forma alternativa que atenda o objetivo desta prática.
- Não satisfaz: se há pontos fracos que comprometem o alcance dos objetivos propostos pela prática.
- Satisfaz Parcialmente: se há pontos fracos, mas estes não comprometem o alcance dos objetivos propostos pela prática.

Para atribuir um nível de satisfação a uma prática, é necessário um julgamento do avaliador. Em avaliações oficiais, profissionais treinados e experientes com o modelo CMMI que atribuem esta classificação. Mas, como estamos falando de pessoas realizando uma atividade de interpretação, há possibilidade de obter-se diagnósticos diferentes se realizados por pessoas diferentes.

Ao final da análise das práticas de cada área de processo, apresenta-se um quadro resumido das práticas e seu nível de satisfação. A seguinte notação foi adotada para expressar o nível de satisfação:

- NS: Não Satisfaz
- S: Satisfaz Parcialmente
- SS: Satisfaz

Para XP ser aderente a determinada área de processo, devemos ter todas as suas práticas com níveis de satisfação S ou SS. Caso a área de processo possua alguma prática com níveis de satisfação NS, esta área será classificada com nível de satisfaz S. E, por fim, se todas as práticas forem classificadas como NS, a área também será classificada como

NS. Esta classificação é válida somente no escopo da análise deste trabalho. Em uma avaliação CMMI, todas as áreas devem ter suas metas alcançadas.

5.2. Mapeamento de Programação Extrema no CMMI nível 2

5.2.1. Mapeamento para a PA Medição e Análise

Objetivo da área de processo

Esta área de processo tem como objetivo fornecer subsídios para o desenvolvimento e manutenção de uma capacidade de medição e análise que será utilizada para dar suporte às necessidades de informação demandadas pela gestão.

Esta área de processo envolve:

- Especificar os objetivos de medição e análise, de forma que estejam alinhados com as necessidades de informação e objetivos identificados.
- Especificar medidas, técnicas de análise e mecanismos para coleta e armazenamento de dados e formas de divulgação e feedback.
- Implementar coleta, armazenamento, análise e relato de dados.
- Fornecer resultados objetivos que possam ser utilizados na tomada de decisões bem fundamentadas e na implementação de ações corretivas apropriadas.

Medição e Análise em XP

Uma das características dos métodos ágeis é exigir um ciclo constante de inspeção, adaptação e melhoria. A escolha das melhores formas de medição é uma tarefa do time completo envolvido no projeto e cabe ao tracker prover informações para a equipe sobre o progresso do time.

Na metodologia Extreme Programming, os objetivos de medição podem ser guiados pelos resultados das reuniões de retrospectiva que encorajam a discussão sobre o processo e trabalho da equipe. Nesta reunião, são destacados os principais pontos de melhoria que a equipe escolheu se concentrar na próxima iteração. A partir destes pontos, o tracker escolhe algumas métricas de acompanhamento para ajudar a equipe a entender e acompanhar o progresso de melhoria em relação aos pontos levantados.

A reunião de retrospectiva é um momento muito importante entre as iterações. É nesta reunião que os resultados da iteração anterior são apresentados pelo tracker, são analisados pelo time completo e pontos de melhoria são identificados. O tracker é responsável por coletar as métricas e apresentar resultados.

Segundo Hartman e Dymond[22], as métricas podem ser classificadas em:

- organizacional ou acompanhamento: As métricas organizacionais tem como objetivo medir a quantidade de valor de negócio entregue ao cliente. Já as métricas de acompanhamento devem ajudar o time no entendimento e melhoria do processo que produz o valor ao negócio.
- objetiva ou subjetiva: Uma métrica objetiva depende somente do objeto medido e não da interpretação de quem a analisa. Já a métrica subjetiva depende do método de análise e também da interpretação de quem está analisando.
- quantitativa ou qualitativa: A métrica quantitativa é expressa por um número ou por um intervalo numérico. Já a qualitativa é representada por palavra ou símbolo, por exemplo, a satisfação do cliente.

Hartman e Dymond[22] propõem um conjunto de verificações que devem ser aplicadas a uma métrica ágil como forma de guiar o tracker para uma melhor escolha:

Lista de verificação de uma métrica ágil	
Característica	**Descrição**
Nome	Não pode ser ambíguo
Classificação	Subjetiva ou Objetiva/ Quantitativa ou Qualitativa/ Organizacional ou de Acompanhamento
Objetivo	Deve indicar uma motivação, preocupação, objeto ou ponto de vista.
Pergunta	Deve estar associada a uma pergunta específica
Base de medição	Uma clara definição das medidas utilizadas para cálculo da métrica
Suposições	Devem ser identificadas para um claro entendimento do que os dados estão representando
Tendência esperada	Uma idéia de qual comportamento esperado para a métrica
Quando utilizar?	Deve esclarecer os motivos que levaram à criação da métrica.
Quando parar de utilizar?	É importante saber até quando uma métrica será útil, antes mesmo de utilizá-la, principalmente métricas de acompanhamento.
Formas de manipulação	Deve esclarecer como as pessoas tentarão alterar seu comportamento em função da métrica para gerar números “mais favoráveis”.
Cuidados e Observações	Recomendações sobre outras métricas similares, limites no uso e perigos associados à má utilização da métrica

Tabela 1 - Adaptado de [21]

Análise de aderência de XP com as práticas de Medição e Análise

SG 1 – Alinhar Atividades de medição e análise

Prática	SP 1.1 - Estabelecer objetivos de medição	Satisfaz
Diagnóstico	XP utiliza os pontos de melhoria apontados pelo próprio time para estabelecer os objetivos de medição. Segundo Hartman e Dymond[22], uma boa métrica ágil deve sempre ter um objetivo associado.	

Prática	SP 1.2 – Especificar medidas	Satisfaz Parcialmente
Diagnóstico	Os pontos de melhoria, apontados nas reuniões de retrospectiva, são utilizados para definir quais os objetivos da medição e como estas serão expressas para o time. Segundo Hartman e Dymond[22], o tracker deve apontar se esta é uma métrica qualitativa ou quantitativa durante o processo de classificação de uma métrica ágil. A classificação de Hartman e Dymond[22] é sugerida e não faz parte das 12 práticas básicas do XP, por isso esta prática é parcialmente satisfeita.	

Prática	SP 1.3 – Especificar procedimentos de coleta e armazenamento de dados	Satisfaz Parcialmente
Diagnóstico	O tracker é o responsável pela coleta e armazenamento dos dados. Uma das práticas básicas do XP, a Semana de 40 horas, é comumente utilizada como intervalo de coleta de métricas durante uma iteração. Uma vez que os objetivos e as medidas foram especificados, cabe ao tracker coletá-los e armazená-los. XP não menciona se os procedimentos de coleta e armazenamento são definidos durante a reunião de retrospectiva ou se é responsabilidade do tracker defini-los.	

Prática	SP 1.4 – Especificar procedimento de análise	Satisfaz
Diagnóstico	Uma prática muito comum em XP é a utilização de pôsteres informativos para divulgar os resultados do trabalho da "Semana de 40 horas". Estes pôsteres também são utilizados para divulgação das métricas semana a semana.	

SG 2 – Fornecer resultados de medição

Prática	SP 2.1 – Coletar dados resultantes de medição	Satisfaz
Diagnóstico	O tracker é o responsável pela coleta das métricas. O intervalo de coleta comumente utilizado é semanal baseado na prática básica de XP "Semana de 40 horas".	

Prática	SP 2.2 – Analisar dados resultantes de medição	Satisfaz
Diagnóstico	O tracker é responsável pela análise dos dados coletados semanalmente. Cabe a ele analisá-los e interpretá-los guiados pelos pontos de melhoria levantados na última reunião de retrospectiva.	

	De acordo Hartman e Dymond[22], uma boa métrica ágil deve ter uma tendência esperada associada. A tendência esperada auxilia na análise dos dados coletados.

Prática	SP 2.3 – Armazenar dados e resultados	Satisfaz Parcialmente
Diagnóstico	O tracker é o responsável pela coleta e armazenamento dos dados. De acordo com a forma de escolhida. Os dados devem ser coletados e armazenados para que sejam comunicados nas reuniões. O método de armazenamento dos dados pode ser gráfico, tabela ou outros elementos de fácil visualização e interpretação. XP não menciona a obrigatoriedade do armazenamento e manutenção dos dados coletados. As métricas não precisam ser obrigatoriamente armazenadas em gráficos ou tabelas. Segundo um dos princípios do Manifesto Ágil[24] é que "o método mais eficiente e eficaz de transmitir informações para e entre uma equipe de desenvolvimento é através de conversa face a face".	

Prática	SP 2.4 - Comunicar Resultados	Satisfaz
Diagnóstico	Um dos princípios do Manifesto Ágil[24] é que o método mais eficiente e eficaz de transmitir informações para e entre uma equipe de desenvolvimento é através de conversa face a face. Outro princípio importante do Manifesto Ágil é Em intervalos regulares, a equipe reflete sobre como se tornar mais eficaz e então refina e ajusta seu comportamento de acordo.	

Tabela resumo do diagnóstico de aderência

Prática	SP 1.1	SP 1.2	SP 1.3	SP 1.4	SP 2.1	SP 2.2	SP 2.3	SP 2.4
Aderência	SS	S	S	SS	SS	SS	S	SS

Tomando os resultados exibidos na tabela acima, podemos afirmar que XP está aderente ou satisfaz a esta área de processo.

5.2.2. Mapeamento para a PA Gerência de Configuração

Objetivo da área de processo

O objetivo desta área de processo é fornecer subsídios para estabelecer e manter a integridade dos produtos de trabalho, utilizando identificação e controle de configuração, balanço das atividades de configuração e auditorias de configuração.

A área de processo de gestão de configuração envolve:

- Identificação da configuração de produtos de trabalho selecionados que compõem os baselines em determinados instantes.
- Controle de mudanças nos itens de configuração.
- Manutenção da integridade dos baselines.
- Disponibilização do status e dos dados atuais de configuração para desenvolvedores, usuários finais e clientes.
- Build de produtos de trabalho a partir do sistema de gestão de configuração ou fornecimento de especificações para fazê-lo.

Os produtos de trabalho que devem ser colocados sobre configuração incluem os produtos internos, que são entregues para o cliente, produtos adquiridos e ferramentas.

Gerência de configuração em XP

A gerência de configuração não está explicita na descrição da metodologia XP. Mas encontramos características da gerência da configuração mencionadas de forma implícita ao longo dos seus valores e práticas.

Kent Beck[25], apresenta como algumas das práticas primárias do XP a propriedade coletiva, a implantação de pequenos releases e a integração contínua. Nestas práticas encontramos características implícitas requeridas pela área de gerência de configuração.

A propriedade coletiva indica implicitamente que o código fonte está acessível a todos os membros da equipe. Além disso, sugere que todos conheçam a sua localização e os procedimentos de acesso, atualização e recuperação deste código. Já as integrações contínuas, após cada tarefa, e a implantação de pequenos releases trabalham com um conceito muito próximo ao conceito de *baseline* da gerência de configuração.

Baseline é um conjunto de especificações ou de produtos de trabalho que tenham sido formalmente revistos, e que, a partir deste ponto serve como base de desenvolvimento ou entrega posterior, só podendo ser modificados por meio de procedimentos de controle de mudança.

Para cada integração, o desenvolvedor deve garantir que tudo está funcionando e todos os testes rodando com sucesso. Os desenvolvedores devem integrar seu trabalho diversas vezes por dia para trabalhar sempre a versão atualizada do sistema. Então a cada integração, tenho um baseline de desenvolvimento que deve servir de base para todos os outros desenvolvimentos seguintes.

Em XP, release é um conjunto de funcionalidades definidas e que representam algum valor para o cliente. Com a prática de *releases curtos*, XP busca entregar software de algum valor para o cliente da forma mais rápida possível. Com entregas incrementais, é possível receber o feedback do cliente ainda durante o desenvolvimento e com tempo

hábil para tomada de ações corretivas. Cada release também pode ser classificado como um baseline de entrega.

Análise de aderência de XP com as práticas de Gerência de Configuração

SG 1 – Estabelecer baselines

Prática	SP 1.1 – Identificar itens de configuração	Não Satisfaz
Diagnóstico	XP não menciona nenhum processo de identificação de itens de configuração. Algumas práticas primárias, apresentadas por Kent Beck[25], sugerem que os itens de configuração sejam compostos pelo código fonte do sistema e que cada item é colocado sobre gestão de configuração no momento de cada integração. Mas o conceito de item de configuração em XP é implícito.	

Prática	SP 1.2 – Estabelecer um sistema de gestão de configuração	Não Satisfaz
Diagnóstico	As práticas de XP não descrevem que é necessário que um sistema de gestão de configuração seja estabelecido. Apesar das práticas de propriedade coletiva e integração contínua sugerirem que há um repositório definido e há procedimentos de armazenamento, atualização e recuperação destes itens, XP não atende outros pontos como: - Mecanismos de gerência para vários níveis de controle; - Criação de relatórios de gestão de configuração; - Estabelecer mecanismos de proteção ao sistema de gestão de configuração(Backups e procedimentos de recuperação); Além disso, XP não menciona procedimentos de configuração para outros itens que não sejam código fonte.		

Prática	SP 1.3 – Criar ou liberar baselines	Satisfaz Parcialmente
Diagnóstico	As práticas de integração contínua e *pequenos releases* implementam a prática de criar e liberar baselines. Porém, XP não menciona nenhum tipo de documentação relacionada a esta baseline contínua que é gerada a cada integração.	

SG 2 – Estabelecer baselines

Prática	SP 2.1 – Acompanhar e controlar mudanças	Satisfaz
Diagnóstico	Para solicitações de mudanças após release, XP considera estes pontos de melhoria ou problemas como uma nova *história* a ser tratada pelo grupo de desenvolvimento. Cada história é documentada em um cartão e discutida, revisada e aprovada por toda a equipe. Após a aprovação desta nova história, esta é estimada e incorporada ao conjunto de funcionalidades do próximo release.	

Prática	SP 2.2 – Controlar itens de configuração	Satisfaz

		Parcialmente
Diagnóstico	Esta prática é parcialmente satisfeita por XP que prevê, em suas práticas primárias, vários procedimentos que são requeridos por esta prática. Porém, XP foca-se apenas no código fonte do sistema e não cita outros itens de configuração produzidos pelo projeto. Em XP, o desenvolvedor deve integrar o seu novo código ao final de cada tarefa. O trabalho em pares garante que o código produzido foi revisado e os procedimentos de integração contínua garantem que o sistema continua funcionando e todos os testes estão rodando após a integração. Isso garante a integridade do build gerado.	

SG 3 – Estabelecer integridade

Prática	SP 3.1 – Estabelecer registros de gestão de configuração	Não Satisfaz
Diagnóstico	XP não menciona nenhuma prática que envolva registros que descrevam mudanças dos itens de configuração.	

Prática	SP 3.2 – Executar auditorias de Configuração	Não Satisfaz
Diagnóstico	XP não menciona nenhuma prática que envolva auditoria dos itens de configuração.	

Tabela resumo do diagnóstico de aderência

Prática	SP 1.1	SP 1.2	SP 1.3	SP 2.1	SP 2.2	SP 3.1	SP 3.2
Aderência	NS	NS	S	SS	S	NS	NS

Tomando os resultados exibidos na tabela acima, podemos afirmar que XP não atende todas as práticas requeridas pela área de processo de gerência de configuração. Assim, esta área será classificada como *satisfaz parcialmente*.

5.2.3. Mapeamento para a PA Gerência de Requisitos

Objetivo da área de processo

O objetivo desta área de processo é fornecer subsídios para gerenciar os requisitos dos produtos e componentes de produto do projeto e identificar inconsistências entre esses requisitos e os planos e produtos de trabalho do projeto.

O processo de gestão de requisitos gerencia todos os requisitos recebidos, técnicos e não técnicos relacionados ao projeto. O projeto deve adotar medidas para assegurar que os requisitos acordados são gerenciados para apoiar as atividades de planejamento e execução do projeto.

Os requisitos recebidos pelo projeto devem ser revisados, juntamente com o seu provedor, para sanar dúvidas e evitar mal-entendidos antes que estes sejam incorporados ao projeto. A partir do acordo entre o provedor de requisitos e responsável por recebê-los, é obtido o comprometimento de ambas as partes com este conjunto de requisitos.

Durante a vida do projeto, os requisitos serão monitorados e suas mudanças ou quaisquer inconsistências geradas pela mudança devem ser gerenciadas. Parte da gestão de requisitos consiste da documentação de mudanças de requisitos e da sua motivação, mantendo rastreabilidade entre os requisitos originais e todos os requisitos de produto e de componentes de produto. A rastreabilidade nos permite identificar facilmente as associações explícitas entre os requisitos.

Gerência de Requisitos em XP

Em XP, os requisitos chegam através das "Histórias" escritas pelo cliente. Cada história descreve uma unidade de funcionalidade que representa um requisito funcional desejado. O cliente deve atribuir uma prioridade para cada história. Esta prioridade será considerada na reunião de planejamento para que o time consiga agregar valor ao negócio do cliente em menor tempo.

Nas reuniões de planejamento, as histórias são discutidas e revisadas pelo time completo. As histórias são estimadas e selecionadas de acordo com a prioridade estabelecida pelo cliente para compor o próximo release. Um dos princípios de XP é a Economia, o time deve preocupar-se com os aspectos econômicos do projeto para que este não seja apenas um sucesso técnico. O projeto deve agregar valor ao negócio do cliente. Por isso que o cliente é o responsável por priorizar suas histórias nas reuniões de planejamento.

A metodologia XP tem como alguns dos seus valores principais a comunicação entre os indivíduos envolvidos e o software em funcionamento. XP pressupõe que a maioria dos problemas, em um projeto de software, estão relacionados a problemas de comunicação. Por isso a prática envolvimento real com o cliente é muito valorizada. Assim, as dúvidas sobre as histórias são sanadas através de comunicação *face-a-face* com o cliente. A colaboração do cliente é fator decisivo de sucesso para um projeto XP.

Outro principio é a Aceitação de responsabilidade. Em XP, a responsabilidade não é imposta e sim, aceita. O desenvolvedor aceita ser responsável por determinada história e também assume a responsabilidade por sua estimativa, implementação e teste.

As mudanças de requisitos, após release, são tratadas como novas histórias em XP. Estas novas histórias entram no fluxo de priorização, análise, revisão e incorporação ao próximo release descrito anteriormente.

Análise de aderência de XP com as práticas de Gerência de Requisitos

SG 1 – Gerenciar Requisitos

Prática	SP 1.1 – Obter entendimento dos requisitos	Satisfaz
Diagnóstico	No jogo do planejamento de um novo release, o time completo analisa e revisa as histórias escritas pelo cliente. Neste momento, as dúvidas são sanadas com o próprio cliente que deve estar presente ou acessível. Umas das práticas de XP, apresentada por Kent Beck[25],é o envolvimento Real com o Cliente. O cliente deve estar acessível a qualquer momento do projeto para tirar dúvidas relacionadas à suas histórias.	

Prática	SP 1.2 – Obter comprometimento com os requisitos	Satisfaz
Diagnóstico	Em XP, a responsabilidade não é imposta e sim, aceita. O fluxo onde o cliente escreve suas histórias e as prioriza, o time revisa e estima e o responsável pela história aceita sua responsabilidade garante o comprometimento de todos os envolvidos com os requisitos.	

Prática	SP 1.3 – Gerenciar mudanças nos requisitos	Satisfaz Parcialmente
Diagnóstico	As mudanças de requisitos, após release, são tratadas como novas histórias. Esta nova história é documentada em um cartão e será priorizada, analisada, revisada e incorporada no próximo release como as demais histórias. Porém, XP não menciona procedimentos de armazenamento destas mudanças e das decisões tomadas sobre cada mudança.	

Prática	SP 1.4 – Manter Rastreabilidade bidirecional dos requisitos	Não Satisfaz
Diagnóstico	XP não menciona nenhuma prática relacionada à rastreabilidade de requisitos.	

Prática	SP 1.5 – Identificar inconsistências entre produtos de trabalho, plano de projeto e requisitos	Satisfaz
Diagnóstico	O desenvolvimento orientado a testes utilizado por XP nos permite identificar rapidamente possíveis inconsistências entre produtos de trabalho e requisitos. Nas reuniões de retrospectiva ou do ciclo semanal, o time completo identifica inconsistências entre o plano de projeto(estabelecido na última reunião de planejamento) e os produtos de trabalho. O plano de projeto de XP é o conjunto de histórias selecionadas para o próximo release.	

Tabela resumo do diagnóstico de aderência

Prática	SP 1.1	SP 1.2	SP 1.3	SP 1.4	SP 1.5
Aderência	SS	SS	S	NS	S

Tomando os resultados exibidos na tabela acima, podemos afirmar que XP atende parcialmente as práticas requeridas pela área de processo de gerência de requisitos.

5.2.4. Mapeamento para a PA Garantia da Qualidade de Processo e Produto

Objetivo da área de processo

O objetivo desta área de processo é fornecer visibilidade para a equipe e gerência sobre os processos e produtos de trabalho. Serve de apoio para a entrega de produtos e serviços de alta qualidade, fornecendo à equipe e gerentes do projeto visibilidade sobre os processos e produtos ao longo do ciclo de vida do projeto.

Esta área de processo envolve atividades de avaliação dos processos, produtos e serviços produzidos pelo projeto em relação às descrições de processo, padrões e procedimentos aplicáveis, identificar e documentar desvios, fornecer feedback à equipe sobre os resultados das atividades de garantia de qualidade e assegurar que não conformidades sejam tratadas.

As avaliações de garantia de qualidade devem ser objetivas. A objetividade é obtida pelo uso de critérios e independência. A independência da área é fator crítico de sucesso para o alcance dos objetivos desta área. As avaliações são feitas por aqueles que não produzem o produto de trabalho.

Em muitas organizações, opta-se pela formação de um grupo de garantia de qualidade que garante a objetividade dos critérios de avaliação. Porém, esta não é a única opção. A garantia da qualidade de processo e produto pode ser executada por pares, total ou parcialmente. Desta forma, a função de garantia de qualidade está embutida no processo.

As não conformidades, encontradas durante o processo de análise, devem ter preferência de tratamento dentro do próprio projeto. Caso esta não possa ser tratada dentro do projeto, deve ser escalada a um nível gerencial apropriado.

Garantia da Qualidade de Processo e Produto em XP

A qualidade em XP pode ser traduzida por um de seus princípios básicos descritos no Manifesto Ágil[24] “Software funcionando é medida primária de progresso”. Porém, este princípio não garante que o software está funcionando de acordo com as especificações do cliente.

A prática primária Programação Pareada de XP prevê que os programadores trabalhem em pares, lado a lado em um mesmo computador. O trabalho em pares durante a execução de uma tarefa promove um trabalho coletivo e colaborativo e melhora a qualidade do código. Os pares não são fixos e devem mudar várias vezes durante o ciclo de vida de um projeto.

Durante a programação pareada, um programador atua como *driver* e o outro como *navigator*. Com esta analogia, o *driver* controla o teclado e tem como atividade

principal produzir as linhas de código necessárias para completar sua tarefa. Já o programador *navigator* atua como um revisor imediato, considerando possíveis riscos de arquitetura do código que está sendo produzido[26].

Esta revisão imediata de XP nos mostra que a garantia do processo de qualidade está embutida no processo. Porém, o programador *navigator* não necessita registrar as não conformidades identificadas. A identificação e tratamento das não conformidades são feitas durante o processo de codificação e de forma exclusivamente verbal.

Análise de aderência de XP com as práticas de Garantia da Qualidade de Processo e Produto

SG 1 – Avaliar Objetivamente Processos e Produtos de Trabalho

Prática	SP 1.1 – Avaliar Objetivamente os Processos	Não Satisfaz
Diagnóstico	XP não menciona práticas ou princípios que nos leva a concluir que esta prática seja atendida. Não há definição de critérios claros para as avaliações de produtos, serviços e produtos de trabalho. Os critérios são definidos pela experiência do programador.	

Prática	SP 1.2 – Avaliar Objetivamente os Produtos de Trabalho e Serviços	Não Satisfaz
Diagnóstico	A avaliação dos produtos de trabalho é feita durante a programação em pares, antes da entrega ao cliente. Porém, não há definição de critérios claros para as avaliações de produtos. Não há processos de identificação e documentação das não conformidades encontradas.	

SG 2 – Fornecer Visibilidade

Prática	SP 2.1 – Comunicar e Assegurar a Solução de Não conformidade	Não Satisfaz
Diagnóstico	A garantia da qualidade proporcionada pela programação em pares não fornece visibilidade das não conformidades encontradas. As não conformidades encontradas são resolvidas pelos próprios membros da equipe. Não há descrição de um procedimento de documentação e monitoramento das não conformidades. Também não há prática ou princípio em XP que descreve a comunicação com o nível gerencial.	

Prática	SP 2.2 – Estabelecer registros	Não Satisfaz
Diagnóstico	Não há descrição de um procedimento de documentação e monitoramento das não conformidades.	

Tabela resumo do diagnóstico de aderência

Prática	SP 1.1	SP 1.2	SP 2.1	SP 2.2
Aderência	NS	NS	NS	NS

Tomando os resultados exibidos na tabela acima, podemos afirmar que XP não atende as práticas requeridas pela área de processo de garantia de qualidade de processo e produto.

5.2.5. Mapeamento para a PA Planejamento de Projeto

Objetivo da área de processo

O planejamento de projeto tem como objetivo estabelecer e manter planos visando definir as atividades do projeto. O plano de projeto fornece a base para execução e controle das atividades do projeto que tratam dos compromissos com os clientes do projeto.

O planejamento inclui estimativa de produtos de trabalho e tarefas, determinação de recursos necessários, negociação de compromissos firmados, elaboração de cronograma e a identificação e análise dos riscos que envolvem o projeto.

O plano de projeto irá fornecer informações que servirão de base para a execução e controle das atividades do projeto. Deverá ser atualizado à medida que o projeto avança para tratar de mudanças de requisitos e processos, ajustes de estimativas e ações corretivas.

Planejamento de Projeto em XP

Um projeto XP se inicia com a fase de planejamento de release, seguido de várias iterações onde cada uma é concluída com o teste de aceitação do usuário. Quando o software tem todas as funcionalidades pedidas pelo usuário naquela iteração, o time finaliza esta iteração e libera o software para ser usado[28].

O cliente escreve e atribui prioridades às histórias que serão analisadas durante o planejamento de release. Na reunião de planejamento, as histórias são discutidas, revisadas, estimadas e selecionadas para compor o próximo release.

Análise de aderência de XP com as práticas de Planejamento de Projeto

SG 1 – Estabelecer Estimativas

Prática	SP 1.1 – Estimar o escopo do projeto.	Não Satisfaz
Diagnóstico	XP não menciona nenhuma atividade que envolva a elaboração de WBS.	

Prática	SP 1.2 – Estabelecer Estimativas para Atributos de Produtos de Trabalho e de Tarefas.	Satisfaz Parcialmente
Diagnóstico	XP adota práticas de estimativa das histórias que compõem um release. Nesta estimativa são considerados os produtos de trabalho e tarefas, abordagem técnica, atributos dos produtos e tarefas e a	

	metodologia XP. A estimativa de um a história é feita utilizando a unidade proposta por Kent Beck[27], *Tempo Ideal*. Porém, esta unidade não define como estimar tamanho para os produtos de trabalho e tarefas. Por isso, a prática é considerada satisfeita parcialmente.

Prática	SP 1.3 – Definir Ciclo de Vida do Projeto	Satisfaz
Diagnóstico	XP possui um ciclo de vida bem definido de seus projetos. As práticas ciclo semanal e trimestral definem os períodos de avaliação e tomada de decisão. Ao final de cada ciclo, a equipe analisa de discute os pontos a melhorar e podem ser realizadas correções de curso do projeto.	

Prática	SP 1.4 – Determinar Estimativas de Esforço e Custo.	Satisfaz Parcialmente
Diagnóstico	A prática 1.2 descreveu como XP realiza suas estimativas. Além disso, XP prevê o uso de modelos ou dados históricos como base das estimativas sempre que estes estiverem disponíveis. A velocidade de uma iteração é a produtividade da equipe e deve ser considerada para as estimativas. Porém XP não menciona como são estimados os custos de infraestrutura de suporte em suas estimativas. Estimativas de viagens, compras de software, treinamentos e equipamentos não são mencionadas.	

SG 2 – Elaborar um Plano de Projeto

Prática	SP 2.1 – Estabelecer orçamento e cronograma.	Satisfaz
Diagnóstico	XP não estabelece um cronograma convencional para o controle de atividades. No planejamento de cada release, XP define datas de início e fim seguindo o ciclo trimestral re seleciona as histórias que farão parte deste release. Como cada história é escrita de forma que represente uma funcionalidade única, isso torna desnecessário o mapeamento das dependências entre elas. Apesar de XP não controlar dependências entre as histórias e não detalhar as atividades e produtos de trabalho que as compõem, isso não fere a prática analisada. Segundo Sommerville[1], a divisão de atividades, que possuam duração menor que uma semana, em unidades menores não é útil para o planejamento. Em XP, usam-se ciclos semanais para planejamento de suas iterações e as tarefas que compõem a iteração possuem duração menor que uma semana.	

Prática	SP 2.2 – Identificar riscos do projeto.	Não Satisfaz
Diagnóstico	XP não menciona procedimentos de identificação de riscos de um projeto.	

Prática	SP 2.3 – Planejar Gestão de dados.	Não Satisfaz
Diagnóstico	Apesar de XP prever um repositório de código unificado em suas	

	práticas, XP não menciona procedimentos para levantamento de requisitos de privacidade e procedimentos de segurança lógica do projeto.

Prática	SP 2.4 – Planejar Recursos do Projeto.	Não Satisfaz
Diagnóstico	XP não menciona nenhuma atividade que envolva a elaboração de WBS, nem seu detalhamento.	

Prática	SP 2.5 – Planejar Habilidades e Conhecimento Necessários.	Não Satisfaz
Diagnóstico	XP não menciona procedimentos para planejamento de treinamentos e avaliação de habilidades e conhecimentos disponíveis.	

Prática	SP 2.6 – Planejar o envolvimento das partes interessadas.	Satisfaz Parcialmente
Diagnóstico	XP estabelece papéis e responsabilidades definidas para todos os stakeholders do projeto. A prática Time Completo prevê que todos devem trabalhar num espírito de colaboração com a equipe. As práticas jogo do planejamento e ciclos trimestral e semanal compõem um cronograma não convencional da interação das partes interessadas. Porém, XP não menciona procedimentos para a documentação destes papéis e responsabilidades, descrição do relacionamento entre as partes, justificativa para o envolvimento de cada parte e recursos necessários para garantir a interação entre os stakeholders.	

Prática	SP 2.7 – Estabelecer o plano de projeto.	Satisfaz Parcialmente
Diagnóstico	Em seu jogo do planejamento, XP define as histórias que farão parte do próximo release de acordo com sua prioridade e estima cada tarefa. Cada história está registrada em um cartão que contém informações como: descrição da tarefa, estimativa e responsável pela tarefa. O conjunto de cartões, das histórias selecionadas, forma o plano de release. Dentro de um release, várias iterações são esperadas, de acordo com a prática Ciclo Semanal. Assim, o conjunto de histórias, que irão compor o ciclo daquela semana, forma o plano de iteração. Porém, nenhum destes planos apresenta descrição de gestão de dados, identificação de riscos, recursos e habilidades e tarefas de gestão.	

SG 3 – Obter comprometimento com o plano

Prática	SP 3.1 – Revisar planos que afetam o projeto.	Não Satisfaz
Diagnóstico	XP não menciona procedimentos de revisão de planos do projeto.	

Prática	SP 3.2 – Conciliar carga de trabalho e recursos.	Satisfaz
Diagnóstico	O princípio Aceitação da Responsabilidade prevê que cada parte interessada aceite sua responsabilidade dentro do projeto. A prática de	

	estimativa de histórias é um bom exemplo. O responsável pela história estima quanto tempo precisará para completá-la. A prática Trabalho energizado prevê que o número de horas dedicadas ao projeto deve ser definido de forma realista. Já a prática Folga, prevê a inclusão de tarefas menores no plano afim de que estas possam ser removidas caso ocorra algum atraso.

Prática	SP 3.3 – Obter comprometimento com o plano.	Não Satisfaz
Diagnóstico	O princípio Aceitação da Responsabilidade prevê que cada parte interessada aceite sua responsabilidade dentro do projeto. Porém, XP não prevê a documentação dos compromissos internos.	

Tabela resumo do diagnóstico de aderência

Prática	SP 1.1	SP 1.2	SP 1.3	SP 1.4	SP 2.1	SP 2.2	SP 2.3
Aderência	NS	S	SS	S	SS	NS	NS
Prática	SP 2.4	SP 2.5	SP 2.6	SP 2.7	SP 3.1	SP 3.2	SP 3.3
Aderência	NS	NS	S	S	NS	SS	NS

Tomando os resultados exibidos na tabela acima, podemos afirmar que XP não atende todas as práticas requeridas pela área de processo de planejamento de projeto.

5.2.6. Mapeamento para a PA Monitoramento e Controle de Projeto

Objetivo da área de processo

O objetivo da área de processo Monitoramento e Controle de Projeto(PMC) é fornecer subsídios para proporcionar visibilidade do progresso do projeto, de forma que ações corretivas apropriadas possam ser implementadas quando o desempenho do projeto desviar-se do plano.

O plano de projeto é o plano global para controle do projeto. É utilizado como base para monitoramento de atividades e implementação de ações corretivas. O progresso do projeto é determinado pela comparação entre o realizado e o planejado de produtos, tarefas, esforço, custo e prazo. Esta comparação é feita em momentos determinados no cronograma do projeto ou em função de níveis definidos no cronograma do projeto ou no WBS. Com a visibilidade que este processo de monitoramento propicia, é possível implementar ações corretivas quando o desempenho do projeto desviar-se significamente do plano. Um desvio é considerado significativo se, quando não resolvido, impede o projeto de alcançar os objetivos estabelecidos.

Monitoramento e Controle de Projeto em XP

A prática Área de Trabalho Informativa de XP prevê que o ambiente de trabalho seja um reflexo do projeto. Ou seja, ao andar pela área de trabalho do projeto devemos ser capazes de ter uma idéia do progresso do projeto, quais são os pontos que estão merecendo mais atenção, qual a produtividade dos programadores, entre outras informações importantes. As métricas também são componentes importantes para compor a área de trabalho informativa.

Análise de aderência de XP com as práticas de Monitoramento e Controle de Projeto

SG 1 – Monitorar o projeto em relação ao plano

Prática	SP 1.1 – Monitorar os parâmetros de planejamento de projeto	Satisfaz Parcialmente
Diagnóstico	Uma das práticas de XP é a área de trabalho informativa. Deve-se ser capaz de ter uma idéia sobre o progresso do projeto apenas andando pela área de trabalho. Com apoio dos procedimentos da área de processo de Medição e Análise, o tracker é responsável pelas tarefas de medição e análise e, através da área de trabalho informativa e as reuniões de ciclo semanal, todas as partes interessadas podem acompanhar o progresso das atividades e da confecção dos produtos de trabalho. Porém, XP não menciona práticas que permitiriam monitorar recursos utilizados, habilidades e conhecimento da equipe e custo e esforço de colaboradores.	

Prática	SP 1.2 – Monitorar compromissos	Satisfaz Parcialmente
Diagnóstico	XP considera as histórias selecionadas como parte do próximo release como um compromisso firmado com o cliente. Cada desenvolvedor deve assumir a responsabilidade pelas suas histórias. Com a ajuda da área de trabalho informativa e das reuniões de ciclo semanal, é possível monitorar os compromissos internos e externos firmados. Porém, XP não prevê a documentação dos resultados das reuniões de ciclo semanal.	

Prática	SP 1.3 – Monitorar riscos do projeto	Não Satisfaz
Diagnóstico	XP não menciona procedimentos que envolvam documentação e monitoramento de riscos do projeto.	

Prática	SP 1.4 – Monitorar a gestão de dados	Não Satisfaz
Diagnóstico	XP não prevê gestão de dados em suas práticas.	

Prática	SP 1.5 – Monitorar o envolvimento das partes interessadas.	Não Satisfaz
Diagnóstico	XP não prevê o monitoramento do envolvimento das partes	

	interessadas como descrito pelo CMMI.

Prática	SP 1.6 – Conduzir revisões de progresso.	Satisfaz
Diagnóstico	Em XP, o tracker é responsável por realizar tarefas de medição e análise e documentar os resultados obtidos. Com a ajuda da área de trabalho informativa e das reuniões de ciclo semanal, é divulgar os resultados obtidos para todos os interessados no projeto.	

Prática	SP 1.7 – Conduzir revisões de marco.	Satisfaz Parcialmente
Diagnóstico	Em XP, as reuniões de ciclo semanal e trimestral podem ser consideradas revisões de marco. Nestas reuniões, compara-se o planejado(histórias planejadas e suas estimativas) com o realizado. Quando são identificados desvios, o plano é revisado e ações corretivas são tomadas. Porém, XP atende parcialmente esta prática, pois não prevê a documentação das questões críticas e seus impactos e nem revisão de riscos do projeto.	

SG 2 – Gerenciar ações corretivas até sua conclusão

Prática	SP 2.1 – Analisar questões críticas	Satisfaz
Diagnóstico	Durante as reuniões de ciclo semanal ou trimestral, possíveis desvios entre o planejado e o realizado e pontos de melhoria são identificados. Cada ponto levantado é discutido a fim de descobrir formas de corrigi-los. A prática Área de Trabalho informativa permite a divulgação das questões críticas e as ações corretivas que foram tomadas.	

Prática	SP 2.2 – Implementar ações corretivas	Satisfaz
Diagnóstico	Os pontos de melhoria são levantados pelo time completo nas reuniões de ciclo semanal ou trimestral. As ações corretivas são indicadas com a participação de todo o time. Caso seja necessário algum replanejamento, a prática Folga permite que algumas histórias possam ser removidas do plano de release e o tempo de folga incluso no plano não atrapalhe a entrega da iteração ou release.	

Prática	SP 2.3 – Gerenciar ações corretivas	Satisfaz
Diagnóstico	As ações corretivas levantadas nas reuniões de ciclo semanal ou trimestral são acompanhadas a fim de determinar sua eficiência. O tracker pode criar um quadro informativo ou novas métricas para monitorar os resultados. Nas reuniões de ciclo semanal ou trimestral, o progresso e eficiência das ações podem ser discutidos e novas ações corretivas podem ser tomadas.	

Tabela resumo do diagnóstico de aderência

Prática	SP 1.1	SP 1.2	SP 1.3	SP 1.4	SP 1.5	SP 1.6	SP 1.7
Aderência	S	S	NS	NS	NS	SS	S
Prática	SP 2.1	SP 2.2	SP 2.3				
Aderência	SS	SS	SS				

Tomando os resultados exibidos na tabela acima, podemos afirmar que XP atende parcialmente as práticas requeridas pela área de processo de Monitoramento e Controle de Projeto.

5.2.7. Síntese da análise das PA´s

A tabela abaixo apresenta um resumo da análise de satisfação obtida em cada área de processo:

Área de Processo	**Nível de Satisfação**
Medição e Análise (MA)	Satisfaz
Gerência da configuração (CM)	Satisfaz Parcialmente
Monitoramento e Controle de Projeto (PMC)	Satisfaz Parcialmente
Planejamento de Projeto (PP)	Satisfaz Parcialmente
Garantia da Qualidade de Processo e Produto (PPQA)	Não Satisfaz
Gestão de requisitos (REQM)	Satisfaz Parcialmente

Os resultados da tabela acima nos mostram que Programação Extrema não é totalmente aderente ao CMMI nível 2.

6. Considerações Finais

O objetivo deste trabalho foi estudar as duas metodologias de desenvolvimento de software e realizar uma análise de compatibilidade entre as duas a fim de que um mesmo ambiente de desenvolvimento pudesse usufruir dos benefícios de ambas.

Com base na análise das práticas das áreas de processo do modelo CMMI nível 2, podemos concluir que XP não satisfaz todas as metas propostas neste nível de maturidade. Porém, com o estudo da metodologia XP apresentado neste trabalho, podemos afirmar que práticas alternativas poderiam ser propostas para que um projeto XP pudesse ser classificado como aderente ao nível 2.

Acredito que o fator determinante que nos levou a concluir que XP não é aderente ao modelo foi a informalidade desta metodologia. XP não valoriza o registro das informações. Não se valoriza a documentação das questões críticas e seus impactos, de

baseline, inconformidades encontradas ao longo do projeto e das reuniões de ciclo semanal ou trimestral.

Além da informalidade de XP, as premissas das duas metodologias são diferentes. O modelo CMMI enfatiza que a qualidade do produto é resultado da qualidade do processo, enquanto XP valoriza as habilidades pessoais e o produto de trabalho.

É importante dizer que o trabalho de análise apresentado neste trabalho foi resultado da interpretação entre as práticas CMMI e XP. Decisões baseadas em interpretações estão sujeitas a um nível maior de contestação do que decisões baseadas em fatos. Este trabalho não afirma que XP não é compatível com CMMI nível 2, e sim que precisa de algumas práticas alternativas para tornar-se totalmente aderente.

Apesar de possuir experiência em projetos de organizações que adotam CMMI nos níveis 3 e 5 e em pequenos projetos onde adotou-se XP, este trabalho seria muito mais rico em sua análise se pudesse ter sido aplicado em um projeto real. Com esta experiência, seria possível levantar problemas reais na adaptação de um projeto XP ao método CMMI.

Este trabalho pode ser estendido a fim de aumentar a sua contribuição para a área de engenharia de software. Um dos trabalhos de extensão sugerido é a análise da compatibilidade XP com CMMI nível 2 em um projeto real com apresentação de práticas alternativas para as metas que não forem alcançadas. Este trabalho resultaria em um guia que poderia ser utilizado por organizações que tem interesse em unir as melhores práticas de ambos modelos.

A adoção de metodologias ágeis entre as organizações aumenta a cada ano. A Programação Extrema é uma boa alternativa principalmente para grupos de desenvolvimento pequenos. Porém, o mercado de software exige alguns processos e procedimentos, presentes nos modelos tradicionais, importantes para formalizar e controlar os projetos que são ignorados. É perfeitamente possível que XP assume estes aspectos importantes sem perder sua agilidade e leveza características e tranqüilizar os clientes mais conservadores.

7. Bibliografia

[1] I. Sommerville, Software Engineering, Fifth Edition, Addison-Wesley, 1995
[2] S.L. Pfleeger, *Engenharia de Software: Teoria e Prática*, São Paulo: Prentice Hall, 2ª edição, 2004. Capítulo 1.
[3] Website Standish Group http://blog.standishgroup.com/ acessado em 10/12/2011
[4] The Standish Group, "Extreme CHAOS 2001", February 2001.
[5] Hillel Glazer, Jeff Dalton, David Anderson, Mike Konrad, Sandy Shrum. CMMI or Agile: Why Not Embrace Both. Technical Note. CMU/SEI, Novembro, 2008.

[6] Dennis Ahern, Aaron Clouse, Richard Turner. CMMI Distilled - A pratical introduction to integrated process improvement, Addison-Wesley, 2007.
[7] Maurício Nacib Pontuschka. Mapeamento entre PMBOK, CMM e RUP. Master´s thesis. 2003.
[8] Geoff Draper, Rick Hefner. An Overview of CMMI Appraisal Methodology. NDIA Systems Engineering Conference, 2001.
[9] Eric Buchholtz, Andy Cordes. Introduction to the Capability Maturity Model Integration(CMMI). RTP SPIN Meeting, 2003.
[10] Alexandre Vasconcelos. Introdução e Experiência de Implantação. Universidade Federal de Pernambuco, 2006.
[11] Gerold Keefer, Hanna Lubecka. The CMMI in 45 Minutes. 2001
[12] SEI Team. CMMI Version 1.2 Overview presentation. 2007.
[13] Suzanne Garcia. Why should you care about CMMI? . Carnegie Mellon University, 2005.
[14] Brian Groarke. The Ten most common excuses for not engaging in process improvement - and what to do about it! .2004
[15] Rushby Craig, Bruce Allgood. CMMI: A comprehensive Overview, 2001
[16] Mike Phillips. CMMI v1.1 Overview. Carnegie Mellon University, 2001.
[17] Mark Schaeffer. DoD Systems Engineering and CMMI, 2004.
[18] CMMI Product Team. CMMI for Development Version 1.3, November 2010.
[19] Mr. Raghav S Nandyal. CMMI Framework of Constellations for Building World-Class IT Organizations, 2011.
[20] Dairton Luiz Bassi Filho. Experiências com desenvolvimento ágil. Master´s thesis, Instituto de Matemática e Estatística da Universidade de São Paulo – IME/USP, Março 2008.
[21] Danilo Toshiaki Sato. Uso Eficaz de métricas em métodos ágeis de desenvolvimento de software. Master´s thesis, Instituto de Matemática e Estatística da Universidade de São Paulo – IME/USP, Agosto 2007.
[22] Deborah Hartmann and Robin Dymond. Appropriate agile measurements: Using metrics and diagnostics to deliver business value. In Agile 2006 conference.
[23] Mark C. Paulk. Extreme Programming from a CMM Perspective. SEI, Novembro de 2001.
[24] http://www.agilemanifesto.org/ acessado em 10/01/2011
[25] Kent Beck and Cynthia Andres. Extreme Programming Explained: Embrace Change. Addison-Wesley Professional, 2nd edition, 2004.
[26] Stuart Wray. How Pair Programming Really Works. IEEE Software January, 2010
[27] Kent Beck and Cynthia Andres. Getting started with XP: Toe dipping, racing dives and cannonballs. Disponível em : http://www.threeriversinstitute.org/ToeDipping.pdf Acessado em 18/01/2012.
[28] Muhammad Javed, Bashir Ahmad, Shahid hussain, Shakeel Ahmad. Mapping The Best Pratices of XP and Project Management: Well defined approach for Project Manager. Journal Of Computing, volume 2, issue 3, Março 2010
[29] Dennis M. Ahern, Aaron Clouse and Richard Turner. CMMI Disttilled: a Pratical Introdution to Integrated Process Improvement, Second Edition. Addison-Wesley, 2008.

Apêndices

Cronograma

O cronograma de trabalho para elaboração da monografia foi divido em semanas e é apresentado abaixo:

Atividades	set/11				out/11				nov/11				dez/11				jan/11			
Estudar o Modelo CMMI v1.3																				
Escrever os capítulos 1 - Introdução e 2- Metodologia																				
Escrever o capítulo 4 sobre CMMI																				
Estudar a metodologia XP segunda edição																				
Escrever o capítulo 4 sobre XP																				
Pesquisar trabalhos de comparação entre CMMI e XP existentes																				
Escrever capítulo de análise e finalização do trabalho																				