



Mapeamento do modelo de Melhoria do Processo de Software Brasileiro (MPS.Br) para empresas que utilizam Extreme Programming (XP) como metodologia de desenvolvimento.

Célio A. Santana, Aline L. Timóteo, Alexandre M. L. Vasconcelos

Centro de Informática – Universidade Federal de Pernambuco (UFPE)
Caixa Postal 7.851 – 50.732-970 – Recife – PE – Brazil

{casj, alt, amlv}@cin.ufpe.br

Abstract. *This paper presents suggestions of changes needed to an enterprise that uses Extreme Programming (XP) as process methodology, it can be certified in level G or F of Melhoria do Processo de Software Brasileiro model (MPS.Br). In this case the enterprise can use the benefits of a disciplined process, easy implementation and fast return of results given by XP and its process appraised by a solid quality model (MPS.Br) at low cost becoming competitive in national market.*

Resumo. *Este artigo apresenta sugestões de modificações necessárias para que uma organização que possua Extreme Programming como metodologia de processo, possa se adequar ao nível G ou F do Modelo de Melhoria do Processo de Software Brasileiro. Neste caso a organização pode se beneficiar de um processo disciplinado, de fácil implantação e de retorno de resultados rápido dados por XP, ao mesmo tempo ter seu processo certificado por um modelo de qualidade consolidado (MPS.Br) a um custo muito baixo tornando-a competitiva no mercado nacional.*

1. Introdução

No início da produção software em meados da década de 1960, não havia nenhuma estrutura formal para desenvolvimento de software. As organizações se deparavam produzindo softwares maiores e mais complexos [Chrissis, Konrad and Shrun 2003] o software deixava de ser um serviço para ser um negócio. Esta maneira *ad hoc* de produzir software levou a uma “crise do software” [Sommerville 1995] uma vez que na maioria dos projetos de software existiam problemas como atrasos e cancelamentos.

A solução encontrada para que a indústria de software melhorasse seus serviços foi à criação da Engenharia de Software. A Engenharia de Software trata de processos, métodos e ferramentas para a produção de software com qualidade. Processos descrevem o que deve ser feito para a entrega efetiva de software, métodos descrevem informações técnicas de como deve ser feito para conceber o software e por fim as ferramentas servem para automatizar processos e métodos [Pressman 1997].

O foco na qualidade é dar suporte e evolução e aprimoramento da engenharia de software. O gerenciamento de qualidade total e filosofias similares fomentam uma cultura de métodos de melhoria contínua de processos e esta cultura leva ao



desenvolvimento de abordagens mais maduras de Engenharia de Software [Pressman 1997].

Um processo de desenvolvimento de software possui o papel de guiar as atividades da equipe, especificar quais e quando os artefatos devem ser gerados, dirigir as atividades individuais a desenvolvedores e à equipe como um todo e oferecer critérios para monitorar e medir as atividades e produtos de projeto [Booch 1995].

As mudanças que estão ocorrendo nos ambientes de negócios têm motivado as empresas a modificar suas estruturas organizacionais e processos produtivos. Alcançar competitividade pela qualidade implica tanto na melhoria de qualidade dos produtos de software e serviços correlatos, como dos processos de produção e distribuição de software. Desta forma, assim como para outros setores, qualidade é fator crítico de sucesso para a indústria de software. Para que este setor no Brasil seja competitivo tanto nacionalmente quanto internacionalmente, é essencial que os empreendedores do setor coloquem a eficiência e eficácia dos seus processos em foco nas empresas [Softex 2005].

Até 2003, segundo dados da Secretaria de Política de Informática e Tecnologia do Ministério da Ciência e Tecnologia (MCT/SEITEC), apenas 30 empresas no Brasil possuíam avaliação SW-CMM®¹ (Capability Maturity Model): 24 no nível 2; 5 no nível 3; 1 no nível 4; e nenhuma no nível 5. Observando-se esta pirâmide pode-se concluir que a qualidade do processo de software no Brasil pode ser dividida em dois tipos de empresas. No topo da pirâmide, normalmente, estão as empresas exportadoras de software e outras grandes empresas que desejam atingir níveis mais altos de maturidade (4 ou 5) do CMMI-SE/SWSM² por estágio e serem formalmente avaliadas pelo SEI (Software Engineering Institute), em um esforço que pode levar de 4 a 10 anos e custar centenas de milhares de dólares. Na base da pirâmide, em geral, encontra-se a grande massa de micro, pequenas e médias empresas de software brasileiras, com poucos recursos e que necessitam melhorar radicalmente seus processos de software em 1 ou 2 anos. Essas empresas precisam saber como adaptar à sua realidade o correspondente aos níveis de maturidade 2 e 3 de modelos para melhoria de processos de software como o CMMI-SE/SWSM e a ISO/IEC 15504-5 [Softex 2005].

O foco principal, embora não exclusivo, do MPS.BR, está neste segundo grupo de empresas. Busca-se que ele seja adequado ao perfil de empresas com diferentes tamanhos e características, públicas e privadas, embora com especial atenção às micro, pequenas e médias empresas, seja compatível com os padrões de qualidade aceitos internacionalmente e que tenha como pressuposto o aproveitamento de toda a competência existente nos padrões e modelos de melhoria de processo já disponíveis. Dessa forma, ele tem como base os requisitos de processos definidos nos modelos de melhoria de processo. O MPS.BR atende a necessidade de implantar os princípios de Engenharia de Software de forma adequada ao contexto das empresas brasileiras, estando em consonância com as principais abordagens internacionais para definição, avaliação e melhoria de processos de software. atende a necessidade de implantar os princípios da engenharia de software de forma adequada ao contexto das empresas brasileiras, estando de acordo com as principais abordagens internacionais para definição, avaliação e melhoria do processo de software [Softex 2005].

¹ © CMM is registered in the U.S. Patent and Trademark Office by Carnegie Mellon University.

² SM CMMI-SE/SW is a service mark of Carnegie Mellon University.



Em fevereiro de 2001 um grupo de dezessete pessoas se uniu em Utah, nos Estados Unidos e assinou o “Manifesto para o desenvolvimento ágil de Software”. Este manifesto tem como primeiro valor definido a seguinte premissa: Indivíduos e Iterações mais do que processos e ferramentas [Manifesto 2001]. A metodologia ágil mais difundida existente é a Extreme Programming (XP) que é alvo deste estudo.

As metodologias ágeis hoje são de interesse de boa parte da comunidade de produção de software, uma vez que suas práticas são bem difundidas principalmente em pequenas configurações. Software mais barato, de mais rápido desenvolvimento, menos burocrático, conformidade com requisitos, melhoria de resultados em curto prazo e implantação barata é um atrativo para pequenas organizações.

As metodologias ágeis pregam que o sucesso de um projeto depende única e exclusivamente das pessoas envolvidas e do esforço por elas realizado para o término do projeto. Enquanto a engenharia de software defende que atos heróicos não acontecerão em todos os projetos e que um processo bem definido pode auxiliar no sucesso da produção de software e repetir este sucesso através de um processo maduro.

Diante das diferenças existentes entre as abordagens do modelo de Melhoria do Processo de Software Brasileiro e Extreme Programming, parece quase impossível que uma organização de menor porte não tenha que fazer uma escolha entre a consistência do primeiro e da agilidade e flexibilidade do segundo.

Mas a questão a ser respondida é: Podem MPS.Br e XP não apenas conviver e coexistir na mesma organização, mas fazer parte de um mesmo processo? Uma vez que MPS.Br se utiliza da formalidade de documentar os processos enquanto XP se denomina leve no aspecto de gerar documentos antes da fase de implementação.

Estas organizações querem se beneficiar das vantagens das metodologias ágeis e também ajustar seu processo com um modelo cuja garantia de qualidade seja comprovada. Mas, ao mesmo tempo, o uso das metodologias ágeis no processo da organização não pode comprometer sua avaliação formal.

2. O que fazer X como fazer (Métodos X Processo)

Algumas sinalizações partem do princípio que XP é um processo disciplinado e documentado, mesmo que com poucos documentos que tem foco na equipe de desenvolvimento e no trabalho técnico, não abordam a institucionalização das práticas de uma organização e se limita pouco a questões relativas à gerência de projetos [Paulk 2001]. XP apenas se opõe a forte cultura de documentação uma vez que mudanças em muitos elementos a tornam mais custosa, portanto ser leve em documentos implica em menos dificuldades para mudanças [Beck 1999].

O MPS.Br tem um foco maior no gerenciamento do processo utilizado pela organização. Não indicando a maneira de como realizar determinadas tarefas, mas indicando o que deve ser feito para que a empresa que se utiliza deste modelo atinja determinado nível de maturidade que sustente seu crescimento e que a mesma seja capaz de repetir bons resultados obtidos em outros projetos anteriores.

Também é observado que XP atende completamente alguns dos requisitos do MPS.Br e a grande maioria das metas de cada processo é atendida parcialmente por XP. O que demonstra uma preocupação comum em entregar software de qualidade, apesar



desta visão de qualidade provir de ângulos diferentes, o que não as excluem mutuamente.

A partir daí podemos perceber que XP tem uma maior tendência a convergir para aquilo que definimos anteriormente como métodos da engenharia de software e o MPS.Br converge para aquilo que foi definido como o processo de desenvolvimento.

3. Extreme Programming (XP)

Kent Beck [Beck 1999] classifica XP da seguinte forma: “*Sistema de práticas que a comunidade de desenvolvedores de software vem evoluindo para resolver os problemas de entregar software de qualidade rapidamente, e então alcançar as necessidades de negócio que sempre mudam*”.

Para tal, além de seguir os quatro valores definidos pelas metodologias ágeis [Manifesto 2001], foi criado um conjunto de práticas, princípios e valores próprios desta metodologia. Também foram identificados alguns fatores hostis a XP em uma organização cujo principal deles é cultura de documentação, ou seja, um ambiente cujo seja necessário à criação de vários documentos antes do início desenvolvimento do sistema [Beck 1999].

Os outros fatores hostis XP, seus princípios e seus quatro valores próprios [Beck 1999], recaem na observação de Paulk [Paulk 2001], que XP tem foco na equipe de desenvolvimento e no trabalho técnico. O importante para este estudo são as práticas de Extreme Programming e como elas afetam a institucionalização do processo.

Algumas das práticas de XP que mais influenciam na definição do processo são as seguintes [Beck 1999]:

- **Jogo do planejamento:** Deve ser elaborado de forma simples, fácil e rápido um plano para o próximo *release*, que consiste em determinar o escopo baseado nas prioridades do negócio, nas possibilidades de implementação e estimativas técnicas;
- **Releases Pequenos:** Os *releases* devem ser de pouca duração e sempre produzir software de valor;
- **Metáforas:** Guiam o desenvolvimento do projeto através de uma estória que explique como o sistema funciona;
- **Testes:** Há dois estágios de testes definidos: testes unitários e testes de aceitação. Ambos devem ser automatizados. Os testes unitários são feitos pelo programador durante o desenvolvimento. Os testes de aceitação são especificados pelo cliente ditando o que deve funcionar para que o software seja aceito;
- **Cliente no Local:** O cliente é parte integrante da equipe e deverá estar presente no local de desenvolvimento sempre que necessário para tirar alguma dúvida ou priorizar alguma estória;
- **Yesterday Wheater:** Estimar ações futuras baseadas em ações iguais ou semelhantes já realizadas no passado para que esta estimativa seja mais precisa.



Extreme Programming também define os papéis da equipe de projeto que são [Beck 1999]:

- **Programador:** É o coração de XP. Responsável por projetar o software, codificar, programar testes e estimar suas tarefas. Todos da equipe assumem este papel no desenvolvimento do projeto;
- **Acompanhador:** É a consciência da equipe XP. Responsável por reportar as métricas do projeto promovendo visibilidade sobre a precisão das estimativas e progresso do projeto;
- **Cliente:** Representante do cliente, responsável por estabelecer prioridades, escopo, escrever histórias e escrever os testes funcionais. Deve estar disponível no local do desenvolvimento sempre que necessário para esclarecer dúvidas;
- **Técnico:** Responsável por identificar problemas e resolvê-los para que a equipe possa trabalhar da melhor forma. Não requer conhecimentos técnicos profundos.
- **Testador:** O testador é responsável por apoiar o cliente a escolher e escrever os testes funcionais, além de assegurar a execução e reportagem dos problemas identificados nestes testes.

O planejamento de projetos em XP foi sendo refinado ao longo de seu desenvolvimento. Primeiro Beck [Beck 2000] define o planejamento de projetos XP com alguns passos que devem ser seguidos para um acompanhamento deste mesmo projeto. Jeffries [Jeffries 2000] também sugere algumas práticas para gerência de projetos que não haviam sido detalhadas pelo trabalho anterior. Em 2002 Thomsett [Thomsett 2002] mostra práticas em gerência de projetos que eram negligenciadas por XP até então como a gerência de riscos abordada nesta referência que até então era pouco explorada em XP.

Outra particularidade de XP é a preferência por software livres e/ou código aberto. Apesar de não ser mencionado esta obrigatoriedade em qualquer referência, sempre encontramos indicações de software livres e código aberto como Wiki e CVS em suas publicações, e as ferramentas concebidas para projetos XP como o XPlanner são em grande maioria neste contexto.

4. Modelo de Melhoria do Processo de Software Brasileiro (MPS.Br)

Em dezembro de 2003 a associação para a promoção da excelência do software brasileiro criou o modelo de Melhoria do Processo de Software Brasileiro. Estes modelos visam o aumento da qualidade dos processos de desenvolvimento de software em organizações que não possuem tantos recursos para investimento em qualidade. Este modelo não define novos conceitos, mas, adequam as estratégias de implementação a realidade brasileira [Webber et al 2004].

A base técnica para a construção do modelo MPS.Br foram as normas ISO/IEC 12207 [ISO/IEC 12207, 1995] e a ISO/IEC 15504 [ISO/IEC 15504] com as quais está em conformidade. Complementarmente, o modelo cobre o conteúdo de outros modelos de referências, como CMMI [Chrissis, Konrad and Shrun 2003] com o qual é compatível. O modelo também define regras para sua implementação e avaliação, de



modo a dar sustentação e garantia de que está sendo empregado de forma coerente com suas definições.

O modelo MPS.Br é composto por três componentes: O Modelo de Referência para melhoria de processo de software (MR-MPS), Método de avaliação para melhoria do processo de software (MA-MPS) e Modelo de negócios para melhoria de processo de software (MN-MPS). Cada componente do modelo foi descrito através de documentos específicos tais como os respectivos guias.

O modelo de referência do MPS.Br é descrito no seu guia geral [MPS BR, 2005]. Nele encontramos definições comuns e o detalhamento do modelo. O guia contém os requisitos organizacionais que devem atender para estar em conformidade com o modelo MPS. O MR-MPS é definido através de sete níveis de maturidade, sequenciais e acumulativos. Cada nível de maturidade é uma junção entre processos e capacidade dos processos, ou seja, é composto por um conjunto de processos em um determinado nível de capacidade [Webber et al 2004].

Os processos e as capacidades dos processos foram descritos segundo as normas ISO/IEC 12207 [ISO/IEC 12207, 1995], e suas emendas 1 e 2, e ISO/IEC 15504-5 [ISO/IEC 15504]. O progresso e o atendimento do nível de maturidade se obtêm, quando são atendidos todos os resultados e propósito do processo; e os atributos de processo relacionados àquele nível e aos anteriores [Webber et al 2004].

Os níveis de maturidade estabelecem patamares de evolução de processos, caracterizando estágios de melhoria de implantação de processos na organização. O MR-MPS define sete níveis de maturidade:

- A (Em otimização);
- B (Gerenciado Quantitativamente);
- C (Definido);
- D (Largamente Definido);
- E (Parcialmente);
- F (Gerenciado);
- G (Parcialmente Gerenciado).

5. Aderência do XP ao MPS.Br (Nível G)

O Nível G do MPS.Br contempla dois processos que são a garantia de requisitos e a gerência de projetos. A gerência de projetos possui quinze práticas a serem cumpridas, enquanto a garantia de requisito possui sete, a tabela abaixo mostra que processos devem ser realizados por nível.

Tabela 1 - Níveis de Maturidade do MR-MPS

Nível MPS.Br	Nível CMMI	Nome e Sigla dos Processos	Atributos de Processo (Capacidade)
A	5	Inovação e Implantação na Organização - IIO	AP 1.1, AP 2.1, AP 2.2, AP 3.1 e



		Análise e Resolução de Causas - ARC	AP 3.2
B	4	Desempenho do Processo Organizacional – DEP Gerência Quantitativa de Projeto - GQP	AP 1.1, AP 2.1, AP 2.2, AP 3.1 e AP 3.2
C	3	Gerência de Riscos - GRI Análise de Decisão e Resolução – ADR	AP 1.1, AP 2.1, AP 2.2, AP 3.1 e AP 3.2
D	3	Desenvolvimento de Requisitos - DRE Solução Técnica – STE Validação - VAL Verificação – VER Integração do Produto - ITP Instalação do Produto – ISP Liberação do Produto – LIP	AP 1.1, AP 2.1, AP 2.2, AP 3.1 e AP 3.2
E	3	Treinamento - TRE Definição do Processo Organizacional – DFP Avaliação e Melhoria do Processo Organizacional - AMP Adaptação do Processo para Gerência de Projeto – APG	AP 1.1, AP 2.1, AP 2.2, AP 3.1 e AP 3.2
F	2	Gerência de Configuração - GCO Garantia de Qualidade – GCA Medição - MED Aquisição – AQU	AP 1.1, AP 2.1, AP 2.2
G	2	Gerência de Projetos - GPR Garantia de Requisitos – GRE	AP 1.1, AP 2.1

5.1 – Gerência de Projetos

O propósito da gerência de projetos é identificar, estabelecer, coordenar e monitorar as atividades, tarefas e recursos que um projeto necessita para produzir um produto e/ou serviço, no contexto dos requisitos e restrições do projeto [Softex 2005]. A gerência de projeto possui as seguintes práticas a serem cumpridas:

- GPR1. O escopo do trabalho para o projeto está definido;
- GPR2. O escopo, os produtos de trabalho e as tarefas do projeto são estimados, através de métodos apropriados;
- GPR3. As fases do ciclo de vida são definidas;



- GPR4. A viabilidade de atingir as metas do projeto, considerando as restrições e os recursos disponíveis, é avaliada;
- GPR5. As tarefas, os recursos e a infra-estrutura necessários para completar o trabalho são planejados;
- GPR6. O cronograma e o orçamento do projeto são estabelecidos e mantidos;
- GPR7. Premissas e restrições são identificadas, incluindo as dependências de tarefas e os critérios para estabelecer ações corretivas;
- GPR8. Os riscos do projeto são identificados e o seu impacto, probabilidade de ocorrência e prioridades de tratamento são determinadas e documentadas;
- GPR9. Os dados relevantes do projeto são identificados, coletados, armazenados e distribuídos. Um mecanismo é estabelecido para acessá-los, inclui questões de privacidade e segurança;
- GPR10. Os recursos humanos para o projeto são planejados considerando o perfil e conhecimento necessário para executar o projeto;
- GPR11. O esforço e custo para produtos de trabalhos e tarefas são estimados baseados em dados históricos ou referências técnicas;
- GPR12. O envolvimento dos interessados do projeto é planejado;
- GPR13. O plano de projeto é revisado com os interessados do projeto e o compromisso com o plano é obtido
- GPR14. Periodicamente o projeto é monitorado com relação ao plano do projeto e o resultado é relatado aos interessados.
- GPR15. Quando há desvio em relação ao plano de projeto do projeto ou quando as metas não são atingidas, são implementadas ações corretivas que são gerenciadas até suas conclusões.

Destas quinze práticas identificamos que XP não satisfaz uma prática (GPR8) e satisfaz parcialmente cinco destas (GPR4, GPR5, GPR6, GPR7 e GPR11). As outras são satisfeitas completamente.

Os problemas identificados neste processo foram:

- Nenhum controle dos recursos utilizados (GPR4 e GPR5);
- Nenhum controle sobre custos e orçamento (GPR6 e GPR 11);
- Não existe dependência entre tarefas em XP (GPR7);
- Nenhum tratamento de riscos é abordado (GPR8).

As soluções sugeridas foram:

- Realizar durante a fase inicial de planejamento (Big Plan) a identificação, avaliação e planejamento de todos os recursos que serão utilizados durante o projeto.
- Criar uma planilha simples que contenham todos os custos previstos com salários, viagens, cursos entre outros e a partir daí criar o orçamento do projeto;



- Criação de uma estrutura de dependência entre as tarefas que são identificadas no projeto;
- Utilização de uma lista de riscos parecida com a de Thomsett [Thomsett 2002], apenas acrescentando as prioridades de tratamento e as probabilidades de ocorrência em cada um deles;
- Utilização da ferramenta CVS para armazenamento de artefatos.

5.2 Garantia de Requisitos

O propósito do processo da garantia de requisitos é gerenciar os requisitos dos produtos e componentes dos produtos do projeto e identificar inconsistências entre estes requisitos e os planos e produtos de trabalho do projeto [Softex 2005]. A Garantia de Requisitos possuem as seguintes práticas a serem cumpridas:

- GRE1. Uma comunicação com o cliente é estabelecida;
- GRE2. O entendimento dos requisitos é obtido;
- GRE3. Critérios objetivos para aceitação dos requisitos são definidos;
- GRE4. O comprometimento com os requisitos é estabelecido, registrado e mantido;
- GRE5. Uma matriz de rastreabilidade bidirecional entre os requisitos, os planos de projetos e produtos de trabalho é gerada e mantida;
- GRE6. Inconsistências entre os planos do projeto, os produtos de trabalho e os requisitos são identificados e corrigidos;
- GRE7. Mudanças nos requisitos são gerenciadas ao longo do projeto.

Destas sete práticas foi identificado que apenas uma prática (GRE5) não era satisfeita por XP, todas as outras seis são completamente atendidas. O único problema identificado foi:

- XP não define nenhuma matriz de rastreabilidade entre requisitos, planos de projetos ou e produtos de trabalho. (GPR 5).

As soluções sugeridas foram:

- Documentar o Big Plan, Planos de Release e Planos de Iteração;
- Atualização do Big Plan quando surgirem novas estórias incorporadas ao sistema;
- Criação de uma matriz de rastreabilidade de estórias;
- Criação de uma tabela de controle de releases, constando quais estórias foram implementadas nele e quais linhas básicas foram geradas no mesmo.

6. Aderência do XP ao MPS.Br (Nível F)

O Nível F do MPS.Br contempla quatro processos que são a gerência de configuração, a garantia da qualidade, medição e aquisição. A gerência de configuração possui quatorze práticas a serem cumpridas, a garantia de qualidade possui quatro, a medição possui sete enquanto a aquisição possui onze.



6.1 Gerência de configuração

O propósito do processo de gerência de configuração é estabelecer e manter a integridade de todos os produtos de trabalho de um processo ou projeto e disponibilizá-los a todos os envolvidos [Softex 2005]. A Gerência de configuração possuem as seguintes práticas a serem cumpridas:

- GCO1. Os itens de configuração são selecionados, segundo critérios documentados;
- GCO2. É especificado o responsável por cada item de configuração e quando o item é colocado sob uma linha básica;
- GCO3. Os itens de configuração que necessitam serem independentemente identificados, armazenados, testados, revisados, usados, alterados entregues e/ou mantidos são identificados, definidos e colocados sob uma linha básica;
- GCO4. É estabelecido um sistema de gerência de configuração que contenha: Múltiplos níveis de controle, mecanismos para armazenamento, retirada e recuperação de itens de configuração e mecanismos para registro de solicitação de mudanças no sistema de gerência de configuração;
- GCO5. O conteúdo do sistema de gerência de configuração é preservado;
- GCO6. Antes de se criar ou liberar as linhas básicas dos itens de configuração é obtida a autorização do grupo de gerência de configuração;
- GCO7. As modificações e liberação dos itens são acompanhadas e controladas;
- GCO8. As modificações e liberações são disponibilizadas para todos os envolvidos;
- GCO9. As situações dos itens de configuração e solicitações de mudanças são registradas, relatadas e seu impacto é analisado;
- GCO10. Revisões para assegurar que mudanças não causem efeitos inesperados nas linhas básicas são executadas;
- GCO11. Os itens de configuração são controlados e itens de configuração alterados só são inseridos no sistema de gerência de configuração após aprovação;
- GCO12. A completeza e consistência dos itens de configuração são asseguradas;
- GCO13. O armazenamento, o manuseio e a liberação dos itens são controlados;
- GCO14. A integridade das linhas básicas é estabelecida e mantida através de auditoria da configuração e dos registros da gerência de configuração.

Destas quatorze práticas foi identificado que apenas três práticas (GCO7, GCO8 e GCO10) são satisfeitas por XP, todas as outras onze XP não satisfaz. Os problemas identificados foram:

- XP não define nenhum critério para seleção de itens de configuração; (GCO1, GCO3, GCO5, GCO11 e GCO12);
- XP não estabelece pessoas envolvidas no processo gerência de configuração (GCO2 e GCO6);
- XP não estabelece sistemas de gerência de configuração (GCO4);
- XP não estabelece análise de impacto (GCO9);
- XP não controla armazenamento e liberações de itens quaisquer (GCO13);
- XP não realiza qualquer auditoria em seu processo (GCO14).

As soluções sugeridas foram:



- Adotar o critério utilizado por alguma referência. Leon [Leon 2000], por exemplo, para identificar itens de configuração;
- Atribuir responsáveis pelos itens de configuração e formar um grupo de configuração;
- Identificar, definir e colocar artefatos sob uma linha básica;
- Utilizar o CVS como sistema de gerência de configuração;
- Verificar o impacto das mudanças e atualizar as planilhas riscos;
- Criar a verificação de completeza e consistências dos itens configuração;
- Substituir as auditorias por testes automáticos.

6.2 Garantia de qualidade

O propósito do processo da garantia de qualidade é garantir que os produtos de trabalho e processos estão em conformidade com os planos e recursos predefinidos [Softex 2005]. A Garantia de qualidade possui as seguintes práticas a serem cumpridas:

- GQA1. A aderência dos produtos e processos aos padrões, procedimentos e requisitos aplicáveis é avaliada objetivamente.
- GQA2. Os produtos de trabalho são avaliados antes de serem entregues ao cliente e em marcos predefinidos ao longo do ciclo de vida de desenvolvimento;
- GQA3. Os problemas e as não conformidades são identificados, registrados e comunicados;
- GQA4. Ações corretivas para não conformidades são estabelecidas e acompanhadas até suas efetivas conclusões.

Destas quatro práticas foi identificado que duas práticas (GQA2 e GQA4) são satisfeitas por XP, as outras duas XP satisfaz parcialmente. Os problemas identificados foram:

- XP não avalia de maneira objetiva a aderência de produtos e processos aos padrões, procedimentos e requisitos aplicáveis (GQA1);
- XP não registra problemas e não conformidades (GQA3).

As soluções sugeridas foram:

- Criar um critério objetivo para verificar a aderência do processo da organização e verificação da corretude da documentação em relação ao processo;
- Criar nota para registrar problemas e mudanças e armazená-la no sistema de gerência de configuração.

6.3 Medição

O propósito do processo de medição é coletar e analisar os dados relativos aos produtos desenvolvidos e aos processos implementados na organização, em seus projetos, de forma a apoiar os objetivos organizacionais [Softex 2005]. A Medição possui as seguintes práticas a serem cumpridas:



- MED1. Objetivos e atividades de medição são estabelecidos a partir das necessidades de informação e objetivos da organização;
- MED2. Um conjunto adequado de medidas, orientado pelas necessidades de informação e objetivos de medição, é identificado e/ou desenvolvido, priorizado, documentado, revisado e atualizado;
- MED3. As atividades de medição (coleta e armazenamento) são especificadas, incluindo-se, métodos e ferramentas;
- MED4. As atividades de análise são especificadas, incluindo-se, métodos e ferramentas;
- MED5. Os dados requeridos são coletados e analisados;
- MED6. Os dados requeridos são armazenados;
- MED7. As informações produzidas são utilizadas para apoiar decisões e para fornecer uma base objetiva para comunicação aos interessados.

Destas sete práticas foram identificadas que duas práticas (MED2 e MED6) são atendidas parcialmente por XP, as outras cinco XP satisfaz completamente. Os problemas identificados foram:

- XP não prioriza as medidas escolhidas. (MED2).
- XP não armazena claramente os dados colhidos (MED6);

As soluções sugeridas foram:

- Criar prioridades para as medidas escolhidas;
- Armazenar os dados colhidos (Os gráficos, medidas de velocidade do programador e etc.) no sistema de gerência de configuração.

6.4 Aquisição

O propósito do processo de aquisição é de se obter um produto e/ou serviço que satisfaça a necessidade expressa pelo cliente [Softex 2005]. A Aquisição possui as seguintes práticas a serem cumpridas:

- AQU1. As necessidades de aquisição, as metas, os critérios de aceitação do produto e/ou serviço e as estratégias (tipos) de aquisição definidas;
- AQU2. Os critérios de seleção do fornecedor são estabelecidos e usados para avaliar os potenciais fornecedores;
- AQU3. O fornecedor é selecionado com base na avaliação das propostas, das capacidades do processo, dos riscos associados e outros fatores;
- AQU4. Um acordo que expresse claramente a expectativa, as responsabilidades e as obrigações de ambos (cliente e fornecedor) é estabelecido e negociado entre o cliente e fornecedor;
- AQU5. Mudanças no acordo, se necessárias, são negociadas entre o adquirente e o fornecedor e documentadas no acordo;
- AQU6. A aquisição é monitorada de forma que as condições especificadas são atendidas, tais como: custo, cronograma e qualidade e, se necessário, ações corretivas são conduzidas;
- AQU7. Revisões, tanto técnicas quanto gerenciais, são conduzidas com fornecedor conforme especificado no acordo;
- AQU8. Informações sobre o progresso técnico são trocadas regularmente com o fornecedor;



AQU9. Procedimentos de aceitação são definidos;

AQU10. O produto e/ou serviço de software entregue é avaliado em relação ao acordado e os resultados da aceitação são documentados;

AQU11. O produto adquirido é inserido no projeto, assegurando-se que as facilidades, treinamento, armazenamento, distribuição e uso do produto adquirido estão de acordo com o que está estabelecido no contrato;

Dentre todos os processos, este é o único que não se aplica a uma organização que segue as tendências da comunidade XP, pois a mesma prioriza softwares livres e/ou código aberto. Esta preferência implica em um outro contrato de licença.

Segundo o guia de aquisição do MPS-Br [Softex 2005b] temos: Considerando ainda que existam modelos de negócio já estabelecidos para o relacionamento cliente e fornecedor de SL/CA, podemos concluir que já existe oferta e demanda por S&SC em SL/CA. Diante do cenário exposto, é importante que as organizações ao considerar a aquisição ou fornecimento de S&SC em um modelo de SL/CA, se preocupem em realizar as devidas adaptações e modificações em seus processos de aquisição, de forma que estes possam acomodar as características particulares desta nova forma de fornecimento e aquisição.

Desta forma temos uma outra forma de licença para negociação destes softwares, portanto, não devem ser sobrescritas. Não é proibitivo que uma organização XP adquira software sob demanda, caso isso ocorra, a empresa deve seguir este processo para se manter aderente ao modelo.

7. Conclusões e trabalhos futuros

Este trabalho foi motivado especialmente pela busca de respostas em relação aos mundos das metodologias ágeis e dos modelos de qualidade. Poderiam eles conviver num mesmo ambiente e, em especial, se seria possível obter benefícios destes dois mundos? Caso isso fosse possível, poderíamos adequá-los ao máximo a realidade brasileira de produção de software? Para isto foram estudados a fundo a metodologia ágil mais conhecida e utilizada atualmente, XP, e os níveis G e F do MPS.Br.

7.1 Principais Contribuições

A primeira grande contribuição deste trabalho se deu na interpretação do atendimento de XP aos níveis G e F do MPS.Br. Este trabalho foi realizado de forma aprofundada, objetivando seguir o mesmo rigor aplicado em uma avaliação oficial do Softex para determinar o nível de maturidade de uma organização. Assim, aqui foram consideradas as práticas do MPS.Br. Em relação à XP, o estudo também se deu de forma aprofundada. Consideramos além das práticas, princípios e valores de XP, aspectos inseridos no dia-a-dia de projetos como as reuniões diárias.

Outra contribuição foi definir um caminho para utilizar XP de forma a estar aderente aos níveis G e F do MPS.Br. Com este guia, empresas que desejem entrega rápida de software e alta produtividade, e que possuem o interesse primordial de seguir modelos de qualidade amplamente reconhecidos, como o MPS.Br poderão utilizar XP em projetos menores, menos críticos, ou em projetos que se espera uma agilidade maior. Juntamente com a diretriz para utilizar XP e MPS.Br nível G ou F em um mesmo ambiente.



7.2 Conclusões

Cada vez mais a comunidade de software vem aceitando, entendendo e aplicando metodologias ágeis. Elas se mostram uma alternativa viável especialmente para grupos e organizações pequenas, por serem mais fáceis de implantar e de ter seus conceitos disseminados por toda a equipe.

Apesar disto, é notório a falta de tratamento de aspectos relevantes do desenvolvimento de software por grande parte destas metodologias. A combinação das metodologias ágeis com estes aspectos seja com base em um modelo de qualidade como o MPS.Br seja com base nos resultados difundidos da engenharia de software, se mostra uma solução factível e viável para diversos ambientes de desenvolvimento.

Além do mais uma metodologia *free* como XP podem ajudar empresas com nenhum processo de software a adquirir a disciplina necessária para produzir um software de qualidade, apoiado em um dos modelos de qualidade reconhecido nacionalmente, a tornando competitiva no mercado nacional.

Referências

- Beck, K. (1999), *Extreme Programming Explained: Embrace Change*, Addison-Wesley.
- Beck, K. (2000), *Planning Extreme Programming*, Addison-Wesley.
- Booch. (1995), *Object Solutions – Managing the Object Oriented Project*, Addison-Wesley, 1995.
- Chrissis, Konrad and Shrun. (2003), *CMMI Guidelines for process integration and product improvement*, Addison-Wesley.
- Jeffries, R. (2000), *Extreme Programming Installed*, Addison-Wesley.
- Leon. (2000), *A Guide to Software Configuration Management*, Artech House.
- Manifesto. (2001) “Manifesto for Agile Software Development”.
<http://www.agilemanifesto.org/> Dezembro.
- Paulk, M. (2001) “Extreme Programming from a CMM perspective”, IEEE, vol. 18 Publishing Press.
- Pressman. (1997), *Software Engineering a practitioner’s approach* 5º Edição, McGraw-Hill.
- Softex. (2005) “Guia geral do MPS.Br V1.0”.
<http://www.softex.br/cgi/cgilua.exe/sys/start.htm?infoid=5723&sid=211>, Dezembro.
- Softex. (2005b) “Guia de Aquisição do MPS.Br V1.0”.
<http://www.softex.br/cgi/cgilua.exe/sys/start.htm?infoid=5722&sid=211>, Dezembro.
- Sommerville, Y. (1995), *Software Engineering* 5º Edição, Addison-Wesley.
- Thomsett. (2002), *Radical Project Management*, Prentice Hall.
- Webber, K. C., Araújo, E., Machado, C., Scalet, D., Salviano, C. and Rocha, A. R. (2004) “Modelo de referência e método de avaliação para melhoria de processo de software”, IV Simpósio Brasileiro de Qualidade de Software.