

# Adoção da Metodologia Extreme Programming para Construção de Software

Thiago M. R. Dias<sup>1</sup>, Jader L. L. Oliveira<sup>1</sup>

<sup>1</sup>CEFET-MG - Centro Federal de Educação Tecnológica de Minas Gerais – Rua Monte Santo – Bairro Santo Antônio 35.500-000 – Divinópolis – MG

thiagomagela@gmail.com, jlamoia@hotmail.com

***Abstract.** This paper deals with the possible use and adoption of the methodology Extreme Programming (XP) in the development of free software communities for developers. As each day comes new communities of free software and this type of development is becoming increasingly popular, feel free to look better if this code developed collectively and consequently its quality. The aim is that as a result this study can provide information for these communities to make the software they developed with more maturity and absorbing the benefits that XP offers.*

***Resumo.** O presente trabalho trata da possível utilização e adoção da metodologia Extreme Programming (XP) no desenvolvimento de software livre por comunidades desenvolvedoras. Como a cada dia surgem novas comunidades de software livre e este tipo de desenvolvimento se torna cada vez mais popular, sente-se um anseio para que se busque uma melhoria neste código desenvolvido coletivamente e conseqüentemente a sua qualidade. O objetivo é que como resultado este trabalho possa fornecer informações para estas comunidades a fim de tornarem o software desenvolvido por elas um software com maior maturidade e absorvendo as vantagens que XP oferece.*

## 1. Introdução

A grande popularização dos software livres, demonstra que esta modalidade de software bem como a sua distribuição representa uma alternativa viável. Desta maneira, surge um anseio para que o software livre, desenvolvido por comunidades se torne cada vez mais confiáveis e com um bom nível de maturidade.

Considerando o crescente surgimento de comunidades que desenvolvem software livre e ainda a grande adoção do pensamento de software livre, que pode ser distribuído e utilizado livremente sem ficar preso a empresas, este tipo de desenvolvimento hoje é altamente aceito e viável.

Por outro lado se tem um histórico que mostra que o software livre tinha um alto índice de fracasso e o não cumprimento de prazos estabelecidos era enorme (Teles, 2005). Outros problemas como erros em excesso e principalmente a falha no levantamento de requisitos fizeram com que os processos de desenvolvimento destes softwares buscassem se adequar a alguma metodologia de desenvolvimento para que os resultados pretendidos pudessem ser alcançados.

O Extreme Programming (XP) apresenta-se como uma metodologia de desenvolvimento ágil que surgiu na década de 90, tende a buscar a qualidade de forma simples a atender as necessidades do cliente, seja ele um cliente externo ou um cliente interno. Este tipo de metodologia tem sido adotada cada vez mais ao contrário das metodologias tradicionais não segue um processo burocrático.

Tendo em vista as necessidades que as comunidades desenvolvedoras de software livre apresentam e as características da metodologia XP, pode-se perceber que algumas das práticas e valores de XP pode ser adotado pelas comunidades a fim de obter um código de qualidade e ter o produto final com mais maturidade.

## **2. Comunidades e Produção de Software Livre**

O software é responsável pela interação entre o usuário e o computador, possibilitando o desenvolvimento de inúmeras tarefas de maneira automatizada. Essa característica é mantida tanto nos softwares básicos, responsáveis por gerenciar o computador, interagindo diretamente com o hardware e fornecendo suporte a execução dos demais programas (exemplo: sistemas operacionais); quanto nos softwares aplicativos, responsáveis pelo desenvolvimento de tarefas específicas no computador (edição de textos, edição de gráficos, planilha eletrônica, planejamento, organização, comunicação, entre outras) (Silveira, 2006).

Segundo definição da Free Software Foundation (FSF, 1991), Software Livre é qualquer programa de computador que pode ser usado, copiado, estudado, modificado e redistribuído sem nenhuma restrição. A liberdade de tais diretrizes é central ao conceito, o qual se opõe ao conceito de software proprietário. A maneira usual de distribuição de software livre é anexar a este uma licença de software livre, e tornar o código fonte do programa disponível.

O termo software livre vem do inglês Free Software e, deriva-se das idéias de Richard Stallman, por meio da FSF. Essa denominação costumou causar certa confusão inicialmente, pelo fato de que a palavra "Free" em inglês pode significar tanto a palavra "Livre" quanto "Gratuito" (Ferraz, 2002). Contudo, o fato de ser livre não significa que não deva ser cobrada uma taxa pela distribuição do software, por exemplo, ou pela disponibilização de materiais que auxiliem na utilização do software (manuais, guias, entre outros), ou ainda, pelo fornecimento de suporte a recursos específicos. A liberdade a que o software livre se refere diz respeito à liberdade de execução, distribuição, estudo e aperfeiçoamento. Sendo, para isso, indispensável o acesso ao código fonte. Estas e outras características são descritas na General Public License (GPL), também referenciada como Licença GNU, da FSF.

De acordo com Alves (2005), percebe-se que o software livre também se encontra inserido em uma ideologia, sustentada principalmente por seu criador Richard Stallman, que prega ideais de livre distribuição e compartilhamento, solidariedade entre os desenvolvedores, entre outros. Alguns membros da comunidade do software livre, não muito satisfeitos com essas características ideológicas imbuídas ao software, iniciaram a partir de 1997 outro movimento, derivado do próprio software livre, denominado Código Aberto (em inglês: Open Source). Aos simpatizantes do movimento Código Aberto, o software livre deve ser utilizado não por questões ideológicas ou sociais, mas sim porque este é o melhor modelo de desenvolvimento de

software com qualidade, capaz de atender de maneira eficiente as mais variadas necessidades de software em diferentes áreas.

Neste trabalho, o termo software livre está sendo utilizado de maneira geral tanto para designar software livre quanto software código aberto.

Também é necessária a distinção entre software livre de software freeware. Neste último, embora seja permitida a livre distribuição e cópia, a liberdade de estudar e de aperfeiçoar o sistema não é garantida e o acesso ao código fonte se restringe necessariamente ao autor do software.

### **3. Desenvolvimento de Software**

O Processo de Software é definido por Sommerville (2007) como um conjunto de atividades e resultados associados que produzem um produto de software. Já Pressman (2006), apresenta a seguinte definição: processo de software é como um framework para as tarefas que são necessárias para a construção de software de alta qualidade.

Na definição de processo de desenvolvimento de software, o mesmo é citado por Vasconcelos (2005) como o conjunto de atividades, métodos, práticas e transformações que são empregados para desenvolver e manter softwares e produtos associados (sendo estes, planos de projeto, documentos de projeto, projetos de software, código, casos de teste e manual do usuário). Atuam neste processo ferramentas e modelos que automatizam e facilitam os trabalhos dos envolvidos (analista, programador, gerente, cliente e outros), possibilitando o desenvolvimento de software com qualidade, obedecendo a prazo e orçamento determinados.

Baseado nestas definições pode-se concluir que não há processo correto ou incorreto; dependendo da sua aplicação, ambiente e objetivo, o uso de um processo específico pode ser vantajoso ou não. Um ponto importante a ressaltar é que, cada autor e organização coloca e classifica processos e atividades de forma diferente, tornando difícil uma uniformidade completa. (Reis, 2001).

O Processo de Software pode ser visto como um gerador de produtos, sendo que o produto final, ou principal, é o Software em si. É importante perceber que existem subprodutos que são gerados para cada fase -- ao final da fase de Especificação, por exemplo, é comum ter sido desenvolvido e entregue um ou mais documentos que detalham os requisitos do sistema.

#### **3.1 Metodologias de Desenvolvimento Ágeis**

Existe discussão até que ponto modelar e especificar um processo é relevante; o certo é que modelos abordam essencialmente questões conceituais da organização do desenvolvimento. Fowler (2001) coloca em que as metodologias modernas de desenvolvimento, como XP e SCRUM, são uma reação a modelos extremamente conceituais e a metodologias monumentais.

Os processos ágeis (ou leves) são em geral empregados por organizações que dão ênfase à colaboração baseada numa abordagem flexível, lidando com projetos nos quais os requisitos mudam constantemente, em decorrência do mercado, da organização, do projeto e do conhecimento. (Silva, 2006).

Na concepção de Beck et al (2001) as principais idéias da Aliança Ágil ressaltam a valorização dos seguintes itens: indivíduos e interações; software funcionando; colaboração com o cliente; e resposta a mudanças. Estes se sobrepõem,

respectivamente, a itens dos processos pesados: processos e ferramentas; documentação abrangente; negociação de contratos; obediência a um plano.

### **3.1.1 Scrum**

Uma metodologia de desenvolvimento que é classificada como ágil é o SCRUM. Esta metodologia foi criada na Easel, e posteriormente desenvolvida por duas empresas em conjunto: Advanced Development Methods e VMARK. Seu objetivo é fornecer um processo conveniente para projeto e desenvolvimento orientado a objeto.

A metodologia é baseada em princípios semelhantes aos de XP: equipes pequenas, requisitos pouco estáveis ou desconhecidos, e iterações curtas para promover visibilidade para o desenvolvimento. No entanto, as dimensões em SCRUM diferem de XP.

SCRUM é interessante porque fornece um mecanismo de informação de status que é atualizado continuamente, e porque utiliza a divisão de tarefas dentro da equipe de forma explícita.

### **3.1.2 Crystal**

Crystal faz parte, na realidade, de um conjunto de metodologias criado por Cockburn (Cockburn, 2004). As premissas apresentadas para a existência deste conjunto são:

- Todo projeto tem necessidades, convenções e uma metodologia diferentes.
- O funcionamento do projeto é influenciado por fatores humanos, e há melhora neste quando os indivíduos produzem melhor.
- Comunicação melhor e lançamentos frequentes reduzem a necessidade de construir produtos intermediários do processo.

Grande parte da metodologia é pouco definida, e segundo o autor, isto é proposital; a idéia de Crystal/Clear é permitir que cada organização implemente as atividades que lhe parecem adequadas, fornecendo um mínimo de suporte útil do ponto de vista de comunicação e documentos.

Atualmente outra metodologia ágil tem ganhado espaço entre equipes que desenvolvem software, esta metodologia chamada de Extreme Programming (XP) será detalhada na próxima seção.

## **4. Extreme Programming - XP**

Extreme Programming (XP) é uma metodologia de desenvolvimento de software, que surgiu nos Estados Unidos na década de 90 seu desenvolvedor foi Kent Beck (Beck, 2004). Como uma metodologia de desenvolvimento ágil, tende a buscar a qualidade de forma simples a atender as necessidades do cliente, seja ele um cliente externo ou um cliente interno.

O interesse por metodologias ágeis como o XP, aparece numa crescente, o que contesta a forma burocrática e rígida mantida por outras metodologias. O XP se apresenta como uma solução viável devido sua fácil adaptação a várias realidades. (Teles, 2004)

Em XP o cliente passa a ser parte integrante da equipe, com papel fundamental de fornecer informações e corrigir possíveis falhas no planejamento e prioridades do software. Na observação de suas práticas é que podemos ver, em que realmente a metodologia do XP auxilia, gestores e desenvolvedores e toda equipe no processo de criação do software. Deixando limpo e claro os caminhos que devem ser percorridos e com isto todos podem trabalhar com o “estado da arte”, para usufruírem de todo seu potencial de criação. (Teles, 2005)

Conforme Tabela 1, a metodologia XP, busca trazer valores e motivações para toda equipe envolvida no projeto de desenvolvimento. Levar confiança ao cliente de que suas necessidades serão respeitadas diante de um informativo de prioridades é de vital importância para que o projeto não seja interrompido.

**Tabela 1 – Valores de Extreme Programming**

Feedback	Comunicação
Simplicidade	Coragem

Mostradas como uma série de atividades a serem seguidas, as práticas de XP norteiam a equipe desenvolvedora durante o projeto. Os valores citados anteriormente, somados as práticas ( Tabela 2), resultam em um emaranhado de boas atitudes.

Existe ampla confiança no entrosamento entre os valores e práticas, de tal forma que os pontos negativos ou fracos de uma são superados pelos pontos fortes de outra.

**Tabela 2 – Práticas de Extreme Programming**

Cliente Presente	Jogo do Planejamento	Integração Contínua
Pequenas Versões	Metáforas	Desenvolvimento Orientado a Testes
Projeto Simples	Time Coeso	Refatoração
Testes de Aceitação	Ritmo Sustentável	Padrões de Codificação
Reuniões em Pé	Posse Coletiva	Programação em Pares

#### 4.1 A Equipe e Seus Membros

Os membros de um projeto em XP, tem papéis que devem ser desempenhados. Como membros de uma equipe temos não apenas os programadores, mas também profissionais responsáveis pelos testes, o gerente que irá cuidar dos recursos necessários. Tabela 3.

É importante perceber que dentro de uma equipe XP, não existem pessoas com ações mais importantes que as outras, não existindo hierarquia. Cada membro faz sua contribuição ao projeto da melhor maneira que for possível.

**Tabela 3 – Membros de Equipe de Extreme Programming**

Gerente de Projeto	Treinador
Analista de Testes	Redator Técnico
Desenvolvedor	Rastreador

## 4.2 Ciclo de Vida

O ciclo de vida em XP é constituído por várias fases que vão sendo vencidas conforme tempo definido anteriormente pela equipe. Existem sugestões de tempo para execução destas etapas, que buscam minimizar problemas com estimativas de tempo e tendem a manter uma constante sustentável no ciclo. São seis as fases que o sistema a ser desenvolvido irá passar, sendo que a velocidade busca completar o ciclo de vida de um projeto em XP em menor espaço de tempo possível. O ciclo de vida curto é propício a projetos em que o sistema sofra mudanças de requisitos frequentemente.

A fase de exploração inicia-se antes mesmo da construção do sistema. O início do ciclo busca resposta para história com possíveis soluções e ainda a possibilidade de implantação destas soluções. Estas experiências são extremamente importantes a equipe para propiciar um primeiro contato com o cliente e ainda ir gerando uma maior confiança a equipe podendo assim reunir um número de histórias satisfatórias a criação de um primeiro release. O tempo sugerido em (Beck, 2001) para esta fase é de duas semanas no máximo.

A fase de planejamento é importante para um acerto de datas entre cliente e equipe desenvolvedora. A execução desta atividade se dá através da definição das histórias que são escritas em cartões pelos clientes. Os programadores fazem suas estimativas de dificuldade e definem prazos para a execução. De posse destes dados podem dizer quais histórias serão implementadas naquela iteração. Os clientes então priorizam as histórias a serem colocadas na iteração. O tempo estimado para esta fase de planejamento é entre dois a quatro meses.

Na fase de iteração do release é que o software vai sendo criado. São escritos os testes funcionais e de unidade. O trabalho é executado seguindo uma ordem de funcionamento em cada iteração. Como produto final desta fase tem-se a apresentação do primeiro release.

A fase de produção começa seu trabalho com os releases criados no sistema, onde estes passam a funcionar em um ambiente similar ao do cliente.

Com a fase de manutenção é que temos a forma mais premente do XP. A todo instante temos a implementação de novas funcionalidades, adicionamos novos membros a equipe e reescrita de código para o seu melhoramento. Estas são as características mais fortes de um projeto em XP.

A fase da morte é o ponto final em um projeto XP, aqui existem duas razões para a morte de um sistema. Uma delas é a conclusão do projeto com aceite do cliente. A outra razão é o projeto ter se tornado inviável para sua continuação, devido a fatores econômicos, e a impossibilidade/dificuldade de se implementar novas funcionalidades com baixo custo, e a uma elevada quantidade de erros no sistema.

## 5. Adoção de Extreme Programming na Produção de Software Livre

Dentre as metodologias apresentadas anteriormente, será abordada a aplicação de Extreme Programming na produção de software livre. É bom ressaltar que materiais que tratam da aplicação de metodologia de desenvolvimento ou processos formais de desenvolvimento por comunidade desenvolvedoras de software livre são escassos, isto porque em sua maioria os processos são pensados para as características de desenvolvimento formal da Engenharia de Software. Isto faz com que possíveis adoções

de metodologias de desenvolvimento para serem utilizadas pelas comunidades, necessitem de adaptações.

A observação deste ambiente permite adaptar as práticas de Extreme Programming às comunidades de software livre. A preocupação inicial que devemos ter é em não restringir o acesso, bem como também não engessar o processo de forma a que este não caminhe, visto que um dos principais motivos do crescimento das comunidades de software livre é a liberdade de escolha (Silveira, 2006).

A escolha da metodologia de Extreme Programming se dá devido a sua capacidade de adaptação, bem como a agilidade criada nos processos de desenvolvimento.

O intuito é apresentar a aplicação da metodologia de XP para as comunidades de software livre devendo sempre manter todo o pensamento e princípios que as comunidades de software livre defendem.

Para o desenvolvimento do projeto, a equipe ou o membro inicial deverá “configurar” a forma de funcionamento da comunidade, assim como as formas de comunicação que serão estabelecidas, papéis que devem ser desenvolvidos por membros da equipe e a forma de divisão destas atividades. Neste ponto deve se destacar a importância da comunicação, um dos valores de XP e que será crucial para todas as fases do projeto.

Sugere-se que a comunicação possa ser feita de forma direta e objetiva, utilizando de todos os recursos hoje disponíveis na internet, como ferramentas VNC e softwares para troca de mensagens instantâneas.

Os membros de um projeto em XP têm papéis, que devem ser desempenhados. Já em comunidades desenvolvedoras a atribuição das atividades deve ser de livre escolha dentro do grupo, o gerente que em suma seria o criador do projeto, teria a responsabilidade de explicitar as atividades a serem cumpridas e os membros vão se inserindo no processo de produção.

O modelo de Bazar descrito por Eric Raymond se enquadra com melhor característica para a execução de refatoramentos, quanto mais gente interessada no código e de posse deste, as correções e adaptações ocorreram mais rapidamente. Código escritos fora dos padrões definidas pela comunidade deverão sofrer refatoração, inicialmente por dois motivos básicos, o primeiro deles é que se coloque o código de acordo com as especificações definidas anteriormente pela equipe e o segundo é gerar um código de fácil leitura principalmente para novos elementos adicionados ao grupo de desenvolvimento (Silveira, 2006).

Já as práticas projetos simples incentivando um código legível, passível de alterações e com um número mínimo de classe e métodos, a posse coletiva que destaca que o código fonte não possui um único proprietário e que todos são responsáveis por todo o programa, bem como padrões de codificação que prega que os padrões devem ser definidos por toda a equipe com regras, a fim de obter um código completamente padronizado apresentam-se como características que estão presente não só no pensamento das comunidades de software livre, mas também em suas atitudes do dia a dia.

## 5.1 Pesquisa Qualitativa sobre o Uso de XP na Produção de Software Livre

Baseado em todas as observações apuradas e discutidas anteriormente, foi realizado um questionário e distribuído na internet para que participantes de comunidades desenvolvedoras de software livre pudessem responder e desta forma ser levantado um perfil destes participantes com relação a sua experiência prática nas comunidades e a possível adoção de XP no desenvolvimento do software livre.

O referido questionário foi disponibilizado em uma página e divulgado em grupos de usuários que são participantes de comunidades de software livre. O questionário bem como os resultados obtidos podem ser encontrados em [www.my3q.com/notice/index.shtml?topic=notice&content=expire](http://www.my3q.com/notice/index.shtml?topic=notice&content=expire).

### 5.1.1 Análise dos Resultados

Após um período destinado para que as pessoas pudessem responder ao questionário, pode-se mapear qual a expectativa e a opinião que as pessoas que participam das comunidades que desenvolvem software livre sentem em relação à possibilidade de adotar a metodologia XP ou parte dela para guiar seus projetos.

O questionário ficou disponível por um período de 90 dias em uma página Web que armazenava todas as respostas dos participantes da pesquisa. No período em que o questionário esteve disponível, o mesmo obteve 153 respostas. Para responder o questionário, instruções no início do mesmo solicitavam que somente pessoas que tivesse um bom conhecimento na disciplina de Engenharia de Software, bem como prática em XP e desenvolvimento de software livre em comunidades respondessem. A análise destas respostas são apresentadas a seguir.

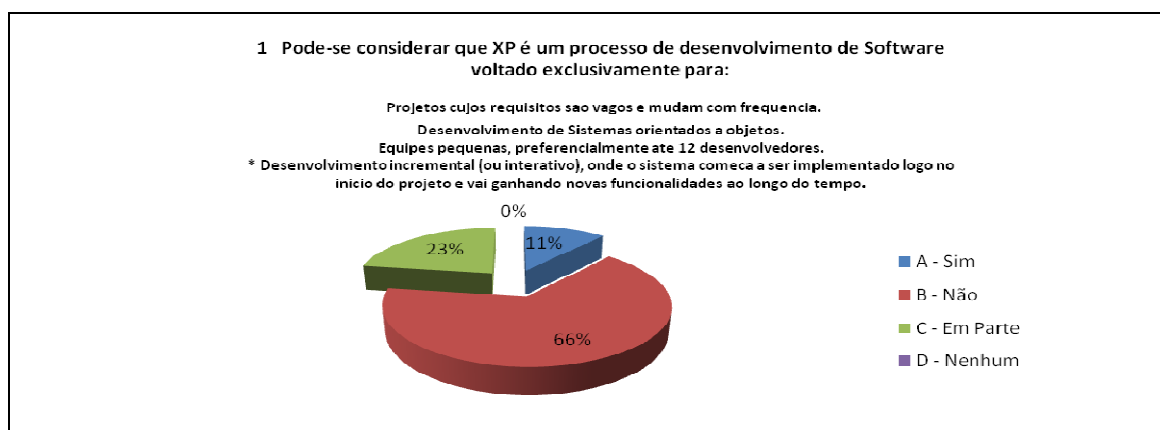


Figura 1 – Questão 01

Baseado nas respostas desta questão pode-se observar que a grande maioria acredita que XP não é uma metodologia de desenvolvimento restrita a uma pequena faixa de projetos e equipes, sendo que outra parte acredita que a metodologia pode em parte ser restrita mas também aplicável a outros tipos de projetos e equipes.



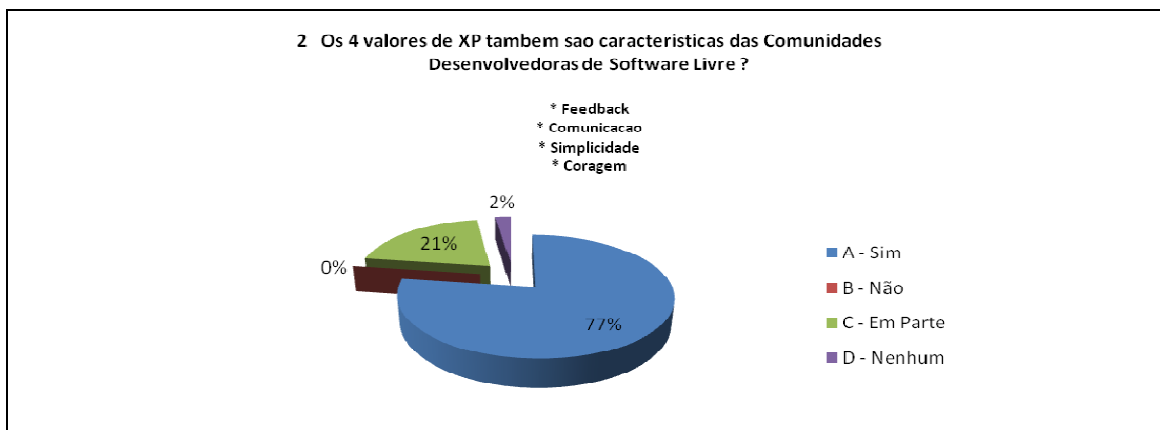


Figura 2 – Questão 02

Em resposta a questão 02 a grande maioria acredita que os valores de XP também são características que estão nas comunidades que desenvolvem software livre. Já outra pequena parcela acredita que em parte pode-se considerar, excluindo desta forma algum valor.

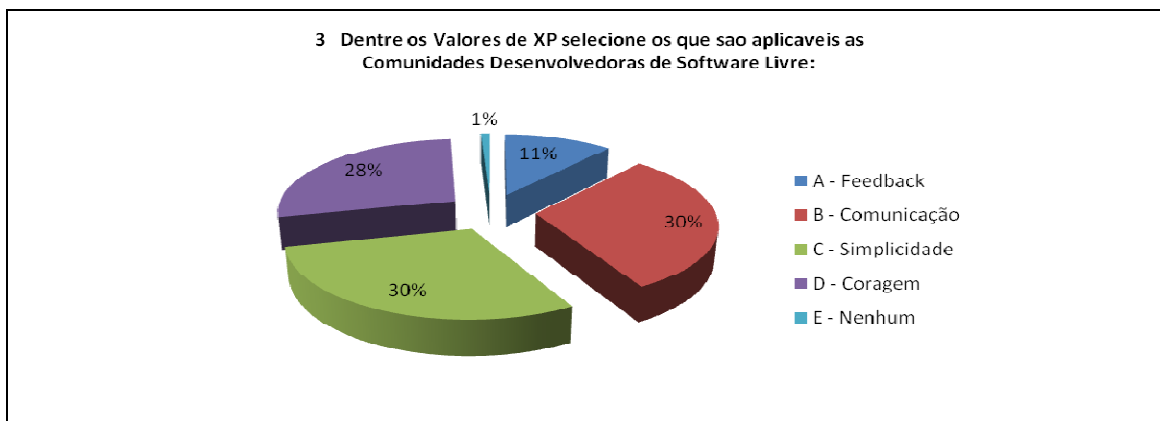


Figura 3– Questão 03

Na questão 03 podemos observar que dentre os valores de XP apenas o Feedback não é citado pela maioria como sendo aplicado às comunidades de software livre, sendo todos os outros valores citados com alto índice de adoção.

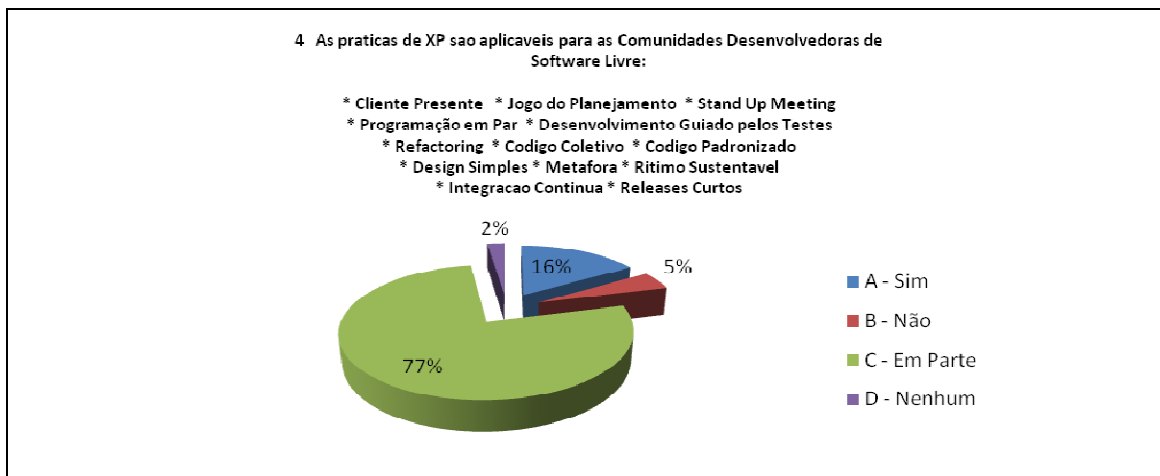
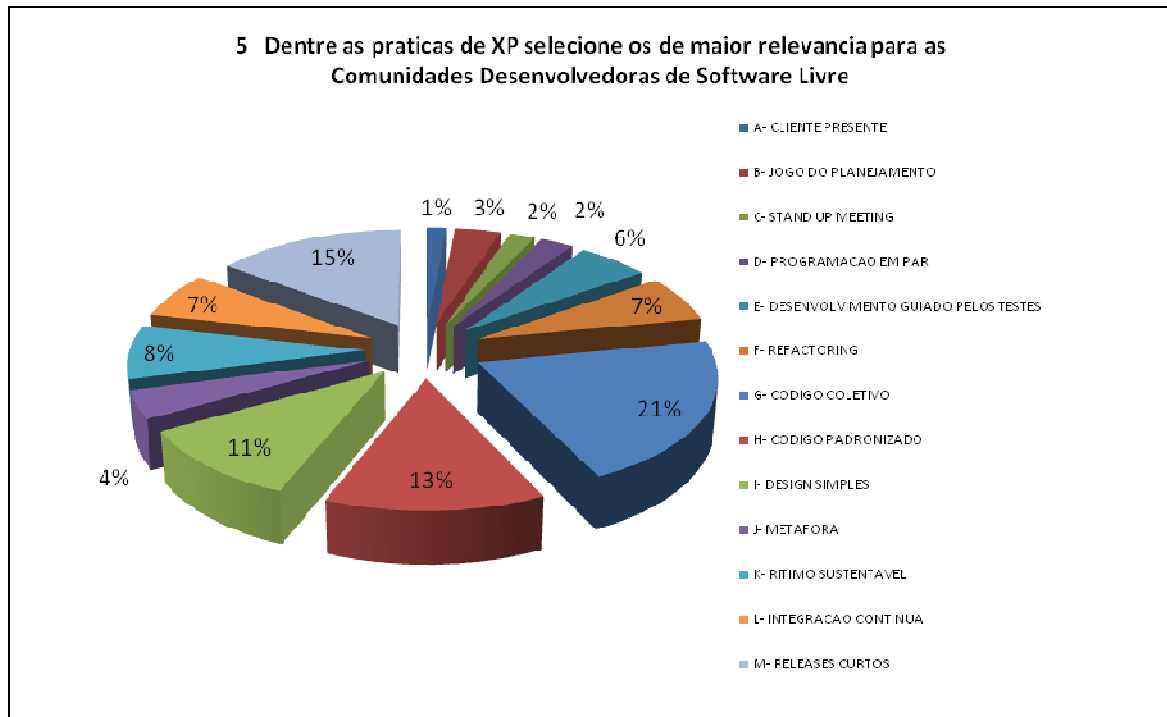


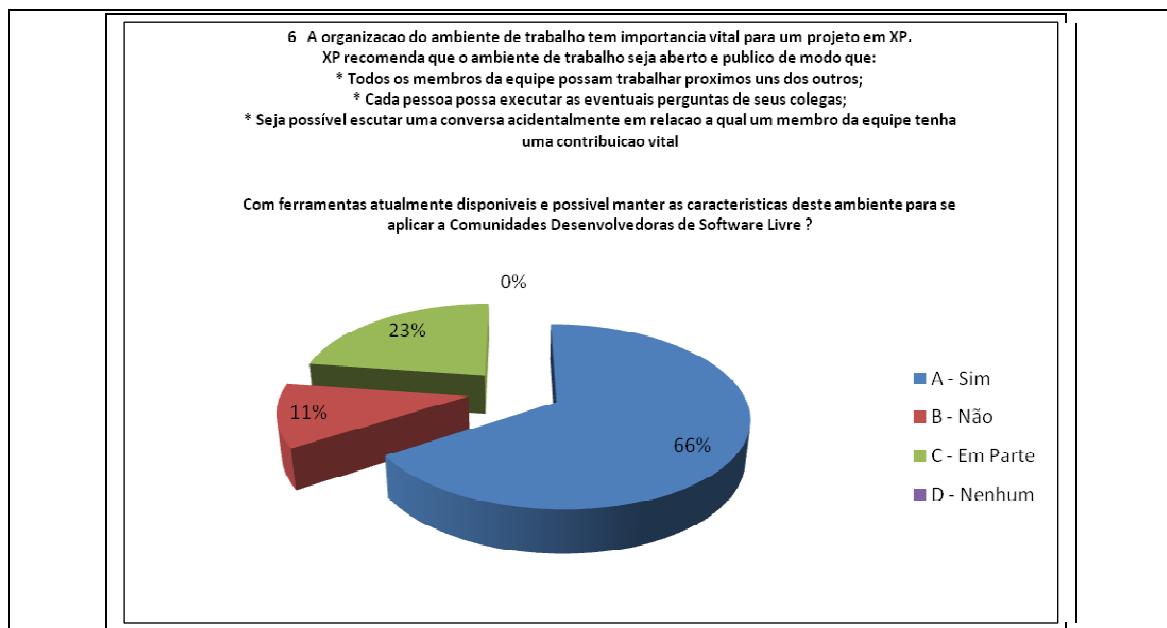
Figura 4 – Questão 04

Baseado nas respostas da quarta questão conclui-se que o maior percentual de respostas apontam que as práticas de XP são em parte aplicáveis para as comunidades de software livre, tendo outra pequena parcela acreditando que elas são em sua totalidade aplicáveis para as comunidades.



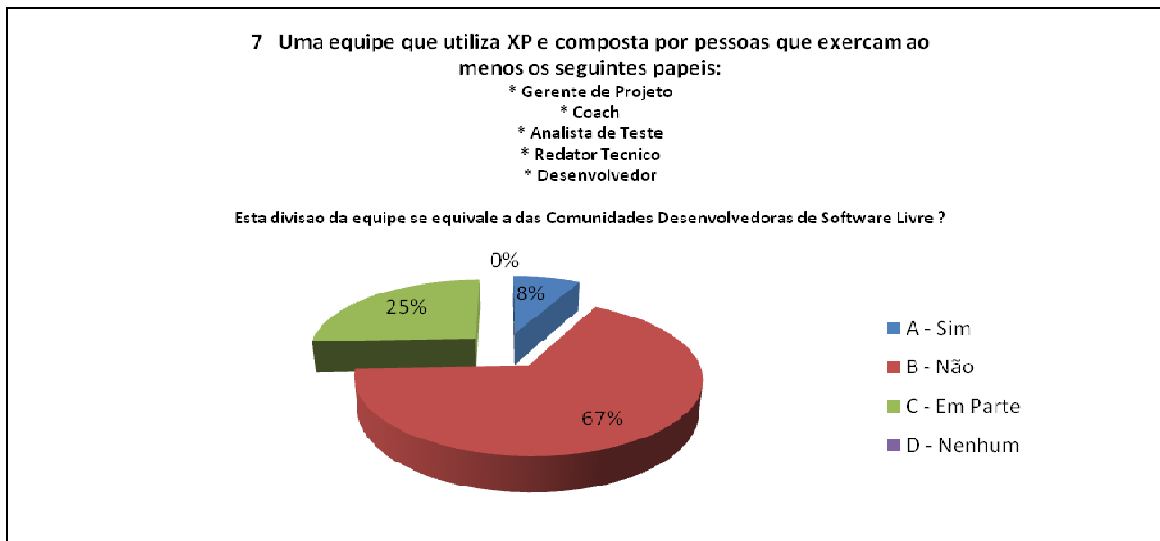
**Figura 5 – Questão 05**

Comparando o resultados sobre a questão cinco, observa-se que dentre as práticas de XP que possuem maior relevância para as comunidades de software livre se destacam o código coletivo, releases curtos, código padronizado, design simples e ritmo sustentável, recebendo as outras práticas um percentual menor de votos.



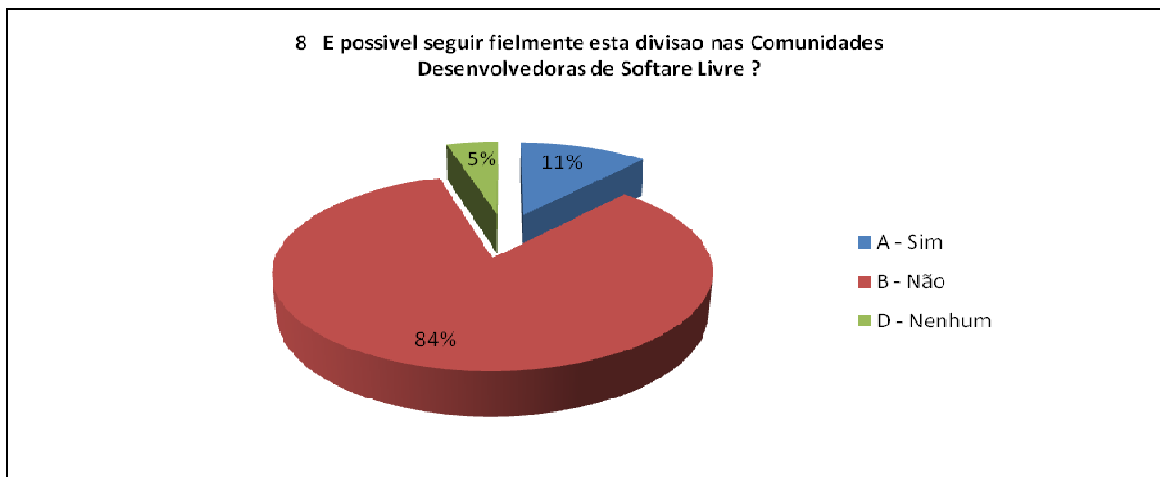
**Figura 6 – Questão 06**

Dentre as recomendações que XP prega para a equipe desenvolvedora de software livre um alto percentual diz que o ambiente de trabalho recomendado pode ser satisfeito se utilizado as ferramentas disponíveis atualmente.



**Figura 7 – Questão 07**

Uma pequena parcela dos participantes do questionário acreditam que a divisão recomendada por XP para compor a equipe de trabalho é compatível com as divisões praticadas nas comunidades desenvolvedoras, acreditando uma parcela que é possível adotar em partes a divisão e a grande acreditando que não se é possível.



**Figura 8 – Questão 08**

Nesta questão onde é enfatizado se a divisão proposta pode ser seguida fielmente o grande percentual de resposta foi a negativa, comprovando a questão anterior onde no mínimo adaptações devem ser feitas para atender as necessidades das comunidades.

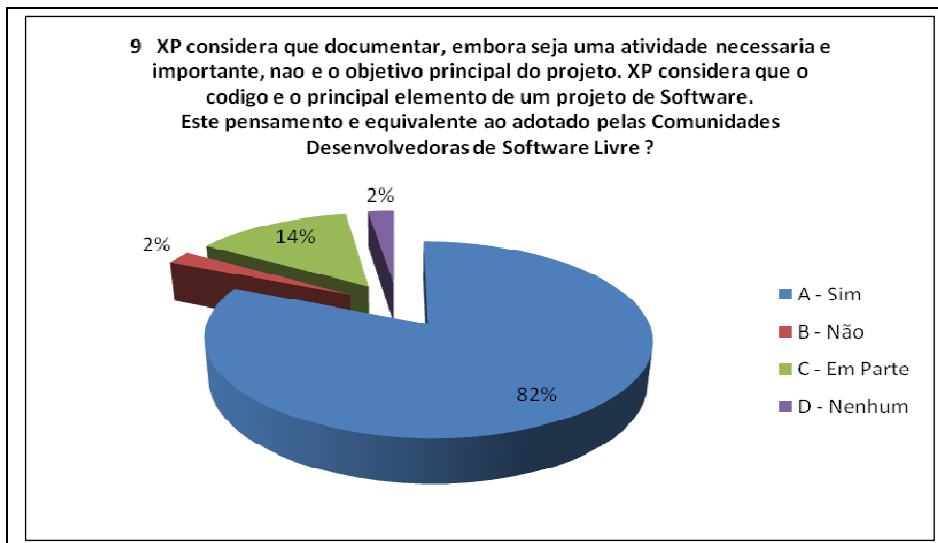


Figura 9 – Questão 09

Com relação a idéia de documentação em XP um pequeno percentual acredita que em parte o pensamento de XP é equivalente aos das comunidades mas a grande maioria acredita que o pensamento de XP com relação a documentação é equivalente a praticada pelas comunidades, ou seja, acreditam que documentar apesar de importante não é o objetivo principal considerando o código como tendo uma grande importância para o projeto.

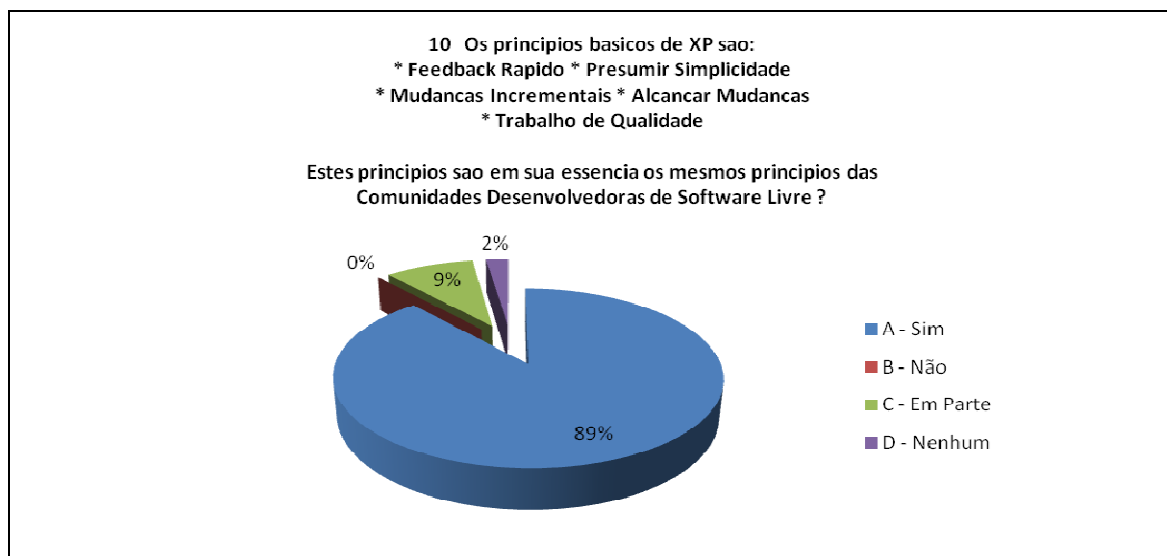


Figura 10 – Questão 10

Com relação aos princípios básicos de XP o grande percentual de entrevistados consideram que eles são na sua totalidade os mesmos princípios das comunidades que desenvolvem software livre.



**Figura 11 – Questão 11**

Finalizando a pesquisa foi questionado sobre qual a metodologia de desenvolvimento entre as mais difundidas, quais mais se adequam as comunidades que desenvolvem software livre e a grande maioria citou o XP, demonstrando desta forma como XP é bem visto e aceito como uma alternativa para as comunidades que desejam adotar uma metodologia de desenvolvimento.

## 6. Conclusões

Os resultados que foram adquiridos e observados no decorrer deste trabalho demonstram que a metodologia XP pode ser aplicada, com adaptações, às comunidades que desenvolvem software livre, se mostrando viável e podendo contribuir para a melhoria dos projetos desenvolvidos.

Pode-se observar que a adoção de metodologias de desenvolvimento por comunidades que desenvolvem software livre ainda é pouco difundida se mostrando grande alternativa para estas comunidades a fim de obter um software com um nível maior de maturidade.

Atualmente com o crescente interesse de pessoas espalhadas por várias partes no desenvolvimento de software livre, o que se tem que buscar é uma maneira de tornar o produto final mais competitivo e alto sustentável. É o que trata este trabalho, levando um conceito sobre a aplicação da metodologia de Extreme Programming para as comunidades de software livre, sem tornar o desenvolvimento algo burocrático.

Aplicar XP nas comunidades necessita que algumas práticas sejam adaptadas e até mesmo algumas sejam ignoradas como é o caso da programação em pares, mas conforme foi apresentado neste trabalho a grande maioria dos valores e práticas de XP são aplicáveis e prometem trazer qualidade para o software desenvolvido.

Baseado nas respostas da pesquisa proposta por pessoas que têm experiência prática como desenvolvimento de software livre, podemos notar que estes vislumbram grande possibilidade de se adotar XP pelas comunidades e apontam grande aplicabilidade ao desenvolvimento.

A elaboração de um estudo baseado no questionário apresentando novas questões e desta forma mapear processos de uma forma mais aprofundada ao desenvolvimento de software livre por comunidades pode ser um trabalho interessante a se desenvolver, já que fica claro a viabilidade de se adotar XP por comunidades desenvolvedoras de software livre.

## Referências Bibliográficas

- Beck, Kent et al. (2001), Manifesto for Agile Software Development. Disponível em: <<http://www.agilemanifesto.org/>>. Acesso em: agosto 2007.
- Beck, Kent (2004), Extreme Programming Explained, Addison Wesley.
- Cockburn, Alistair. (2004), Crystal Clear, Addison Wesley.
- Ferraz, Nelson C. de Toledo. (2002) Vantagens estratégicas do software livre para o ambiente corporativo. Monografia apresentada para conclusão do curso de Master Busines Information Systems. São Paulo: MBIS / PUC-SP.
- Fowler, M. (2001), The New Methodology. Disponível em <http://www.martinfowler.com/articles/newMethodology.html>. Acesso em agosto 2007.
- FSF, FREE SOFTWARE FOUNDATION. (1991) License GNU General Public License. Boston: FSF. Disponível em: <<http://www.fsf.org/licenses/gpl.html>>. Acesso em: agosto 2007.
- Pressman, Roger S. (2006), Engenharia de Software. 6ª. ed. São Paulo. McGraw-Hill.
- Reis, Christian (2001), Caracterização de um Modelo de Processo para Projetos de Software Livre. Dissertação de Mestrado. USP, São Carlos. SP.
- Silva, Fernando S. (2006), Os Processos de Desenvolvimento do Software Livre: Um Estudo Exploratório. UFLA/FAEPE, Lavras.
- Silveira, Daniel R. (2006); Viabilidade do uso de Extreme Programming em ambientes de desenvolvimento livre, com grupos de desenvolvimento alocados em espaços geográficos diferentes. UFLA/FAEPE, Lavras.
- Sommerville, I. (2007), *Software engineering*. 8th. ed. Addison-Wesley.
- Teles, Vinícius M. (2004), Extreme Programming. São Paulo: Novatec.
- Teles, Vinícius M. (2005), Um Estudo de Caso da Adoção das Práticas e Valores do Extreme Programming. Dissertação de Mestrado, IM-NCE/UFRJ, Rio de Janeiro.
- Vasconcelos, Alexandre M. L. (2005), Produção de Software (com ênfase em Software Livre): processos de desenvolvimento de software 1. Lavras: UFLA/FAEPE.