

CENTRO PAULA SOUZA

GOVERNO DO ESTADO DE
SÃO PAULO

**Faculdade de Tecnologia de Americana
Curso Superior de Tecnologia em Análise e Desenvolvimento de
Sistemas**

***EXTREME PROGRAMMING: METODOLOGIA
ÁGIL***

GLAUCIA SCHNOELLER DOS SANTOS

Americana, SP
2013

CENTRO PAULA SOUZA

GOVERNO DO ESTADO DE
SÃO PAULO

**Faculdade de Tecnologia de Americana
Curso Superior de Tecnologia em Análise e Desenvolvimento de
Sistemas**

EXTREME PROGRAMMING: METODOLOGIA ÁGIL

GLAUCIA SCHNOELLER DOS SANTOS
glaucya.santos@hotmail.com

Trabalho Monográfico, desenvolvido em cumprimento à exigência curricular do Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas da Fatec-Americana, sob orientação do Prof. Anderson Luiz Barbosa.

Área: Engenharia de software

**Americana, SP
2013**

BANCA EXAMINADORA

**Professor: Anderson Luiz Barbosa
(Orientador)**

Professor: Alberto Martins Júnior

Professora: Maria Elizete Luz Saes

AGRADECIMENTOS

Agradeço primeiramente a Deus, por me iluminar e me dar força em cada etapa presente na minha vida.

Agradeço aos meus pais, Roseli S. Santos e Diogenes S. Santos, pela compreensão e apoio na realização deste curso.

Aos meus amigos, em especial aos meus queridos parceiros de equipe, Samanta Branquinho, Virginia Trinca, Pamela Chagas, Suelen Magnaterra e Douglas Araujo, pela fraternidade e companheirismo que levarei para sempre comigo.

Aos professores que me auxiliaram no decorrer do curso, ao meu orientador Anderson Luiz Barbosa com o auxílio no desenvolvimento deste trabalho.

DEDICATÓRIA

Dedico este trabalho a Deus, as meus pais, Roseli Santos e Diogenes Santos, pelo incentivo e paciência, aos meus amigos, pelo companheirismo durante o curso e aos meus professores pelo ensinamento adquirido.

RESUMO

A metodologia ágil *Extreme Programming*, visa desenvolver sistemas em curto prazo e com excelente desempenho. A XP é baseada em um conjunto de valores, princípios e práticas que auxiliam no projeto de desenvolvimento de sistema. Além disso, a metodologia segue um ciclo de vida, que são executadas algumas etapas contínuas. A XP oferece diversas vantagens, como adaptabilidade as mudanças constantes, interação do cliente, ambiente saudável de trabalho, entre outros. Esta metodologia é muito conhecida e utilizada.

Este trabalho propõe o estudo detalhado desta metodologia, objetivando apresentar um mecanismo eficaz para se obter sucessos nos projetos de desenvolvimento de sistemas. Para elaboração deste trabalho foi realizada pesquisas bibliográficas e documentais, e também um estudo de caso da implantação da XP em um projeto empresarial.

Palavras Chave: *Extreme Programming*, Metodologia ágil, XP.

ABSTRACT

The agile methodology Extreme Programming aims to develop systems with short term and with excellent performance. The XP is based on a set of values, principles and practices that assist in system development project. Furthermore, the method follows a life cycle that some steps are performed continuously. The XP offers several advantages such as adaptability to constant change, customer interaction, healthy work environment, among others. This Methodology is well known and used.

This paper proposes a detailed study of this methodology, aiming at presenting an effective mechanism to achieve success in systems development projects. For preparation of this work was performed research literature and documents, and also a case study of the deployment of XP on a business project.

Keywords: Extreme Programming, Agile methodology, XP.

ÍNDICE DE IMAGENS

Figura 1: Ciclo de um <i>release</i> em <i>Extreme Programming</i>	18
Figura 2: Cartão de história.....	29
Figura 3: Cartão de tarefa.....	30
Figura 3: Fluxo de atividades.....	36

SUMÁRIO

1. INTRODUÇÃO	9
1.1. JUSTIFICATIVA	11
1.2. PROBLEMA.....	11
1.2.1. PERGUNTA	11
1.3. HIPÓTESES	11
1.4. OBJETIVOS	12
1.4.1. OBJETIVOS ESPECÍFICOS	12
1.5. METODOLOGIA.....	12
2. METODOLOGIA ÁGIL E <i>EXTREME PROGRAMMING</i>.....	14
2.1. INTRODUÇÃO DA METODOLOGIA ÁGIL	14
2.2. <i>EXTREME PROGRAMMING</i>	16
2.2.1 VANTAGENS E RESTRIÇÕES	18
3. VALORES E PRINCÍPIOS BÁSICOS DA XP.....	21
3.1. VALORES	21
3.2. PRINCÍPIOS BÁSICOS	22
4. PRÁTICAS DA XP	29
4.1. CLIENTE DISPONÍVEL/PRESENTE.....	29
4.2. JOGO DE PLANEJAMENTO	29
4.3. <i>STAND UP MEETING</i>	30
4.4. PROGRAMAÇÃO EM PAR/DUPLA	30
4.5. <i>REFACTORING (REFATORAÇÃO)</i>	31
4.6. DESENVOLVIMENTO GUIADO POR TESTES.....	31
4.7. CÓDIGO COLETIVO	31
4.8. PADRÕES DE DESENVOLVIMENTO.....	31
4.9. DESIGN SIMPLES	31
4.10. METÁFORA	32
4.11. RITMO SUSTENTÁVEL	32
4.12. INTEGRAÇÃO CONTÍNUA	32
4.13. <i>RELEASES CURTOS</i>	32

5.	EQUIPE XP.....	33
5.1.	GERENTE DE PROJETO.....	33
5.2.	<i>COACH</i>	33
5.3.	ANALISTA DE TESTE	33
5.4.	REDATOR TÉCNICO.....	344
5.5.	DESENVOLVEDOR.....	34
6.	CICLO DE VIDA DO PROJETO XP	35
6.1.	FASE DE EXPLORAÇÃO	35
6.2.	FASE DE PLANEJAMENTO INICIAL	35
6.3.	FASE DE ITERAÇÕES DO <i>RELEASE</i>	35
6.4.	FASE DE PRODUÇÃO.....	36
6.5.	FASE DE MANUTENÇÃO.....	36
6.6.	FASE DE ENCERRAMENTO DO PROJETO	36
7.	ESTUDO DE CASO: IMPLANTAÇÃO DA <i>EXTREME PROGRAMMING</i> NA EMPRESA ANCAR.....	37
8.	CONSIDERAÇÕES FINAIS.....	43
	REFERÊNCIAS	46

1. INTRODUÇÃO

Para Ambler (2004) um projeto de desenvolvimento de sistema deve resultar em um software que atenda as necessidades do cliente de forma eficaz. Porém a atual situação de desenvolvimento de sistema está abaixo do ideal.

Devido ao ambiente globalizado em que vivemos, estamos sujeitos a rápidas mudanças no mercado e assim conseqüentemente influenciando no desenvolvimento de software. Com as mudanças frequentes, os requisitos se tornam vulneráveis e estão sempre sendo modificados. Por esta razão o percentual de projetos que atingem as metas propostas e conseguem a satisfação do cliente, representam uma pequena quantidade do total de sistemas desenvolvidos. Os sistemas devem ser desenvolvidos rapidamente para atender as novas oportunidades.

Os frequentes problemas ocorridos nos projetos de desenvolvimento são:

- Alteração no prazo do cronograma;
- Funcionalidades que não atendem as necessidades do usuário;
- Baixa eficiência do sistema;
- Ajuste no orçamento do projeto.
- Equipe exausta;
- Insatisfação dos clientes;

Diante deste cenário surgiu a necessidade de desenvolver um sistema mais flexível e dinâmico, com o intuito de corrigir estas falhas foi desenvolvida a metodologia ágil. Indicativos mostram que a metodologia ágil mais utilizada e conhecida é a *Extreme Programming*, que será o tema principal desta monografia.

O primeiro projeto com a *Extreme Programming* teve início no dia seis de março de mil novecentos e noventa e seis, e veio se afirmando como uma das metodologias mais utilizadas, obtendo sucesso em empresas de diferentes tamanhos e atividades de negócios. É composta por práticas e valores que asseguram ao cliente um software funcional e adaptável a mudanças, tornando o processo mais produtivo, diminuindo o tempo e os custos do projeto. Entretanto, deve ser aplicado em projetos de equipes pequenas e médias, onde os requisitos

não necessitam ser analisados detalhadamente, e que sejam vagos e em mudanças constantes. Desta maneira, é possível equilibrar a qualidade e o comprometimento dos requisitos do software.

A *Extreme Programming* se tornou conhecida pela simplicidade oferecida, pois não segue uma sequência rígida de documentação, é focada na parte de desenvolvimento. Esta metodologia apresenta inúmeros benefícios ao projeto, tornando o processo de desenvolvimento mais ágil e com um maior desempenho da equipe.

ESTRUTURA DO TRABALHO

O trabalho foi estruturado em oito capítulos, sendo que o primeiro traz a descrição da justificativa, do problema, das hipóteses, dos objetivos para o estudo e a metodologia utilizada. O segundo capítulo apresenta a introdução da metodologia ágil, abordando a formação e os conceitos desta metodologia. E descreve o conceito da metodologia ágil *Extreme Programming*, enfatizando suas vantagens e restrições na implantação. No terceiro capítulo são esclarecidos os valores e princípios básicos da XP e as melhorias com a implementação e utilização em um projeto de desenvolvimento. No quarto capítulo é demonstrado o estudo das práticas que compõe a metodologia XP, e a aplicação dos valores e princípios básicos durante o desenvolvimento do sistema. O quinto capítulo é composto pela a formação e papel da equipe da metodologia *Extreme Programming*. O sexto capítulo descreve cada fase de um ciclo de vida da metodologia *Extreme Programming* em um projeto. Para demonstrar a prática da implementação o sétimo capítulo traz o estudo de caso realizado na empresa Ancar.

Com base no estudo realizado nos capítulos anteriores, o oitavo capítulo se reserva as considerações finais do trabalho.

1.1. Justificativa

Para tanto o estudo se justificou pela importância em apresentar uma forma eficaz para melhorar a produtividade e o desenvolvimento nos projetos de software, objetivando um aumento no desempenho do sistema, a satisfação do cliente e da equipe, agilizando e disponibilizando a elaboração e a entrega do projeto.

Aplicando – se essas práticas da *Extreme Programming* o cliente tem um *feedback* contínuo, tendo uma participação interativa com a equipe e assim resultando em uma melhoria no entendimento das necessidades do cliente. Essas práticas de acordo com Ambler (2004) “[...] é uma maneira de trabalhar em conjunto de modo eficaz para alcançar os objetivos dos clientes do projeto.” A motivação para a escolha deste tema surgiu da necessidade de oferecer sistemas de qualidade aos clientes.

1.2. Problema

O problema abordado foi: Através de estudos é demonstrado que grande parte dos projetos de software falha no cronograma, no prazo, ou fatores que não atendem às requisições dos clientes.

1.2.1. Pergunta

Como pergunta que se buscou responder: A utilização da metodologia ágil *Extreme Programming* aumenta a eficiência e qualidade no projeto de desenvolvimento de um sistema?

1.3. Hipóteses

As Hipóteses apresentadas foram: Com a utilização da metodologia *Extreme Programming*, há uma facilidade na adequação de mudanças no projeto melhorando a produtividade. O uso da metodologia ágil *Extreme Programming*, traz um retorno contínuo para o cliente, porém, a equipe pode correr o risco de ter que “reprojetar” o sistema. Para projetos de sistemas críticos que exijam uma documentação completa do sistema, a metodologia *Extreme Programming* não deve ser aplicada.

1.4. Objetivos

O objetivo geral deste estudo foi pesquisar os conceitos e valores da metodologia ágil *Extreme Programming*, buscando a facilidade no desenvolvimento de sistemas.

1.4.1. Objetivos específicos

Os objetivos específicos foram: Fazer um levantamento bibliográfico sobre a metodologia ágil *Extreme Programming*, objetivando o entendimento da utilização e aplicação desta metodologia. Descrever os conceitos e valores da metodologia estudada, visando esclarecer as etapas necessárias para o desenvolvimento de um sistema e, verificar se os conceitos foram esclarecidos, buscando apresentar a metodologia ágil *Extreme Programming* como uma solução para o desenvolvimento de software.

1.5. Metodologia

Para a elaboração deste trabalho, foi adotada a pesquisa bibliográfica, a pesquisa documental e o estudo de caso. Para Severino (2007, p. 122) a pesquisa bibliográfica é:

Aquela que se realiza a partir do registro disponível, decorrente de pesquisas anteriores, em documentos impressos, como livros, artigos, teses etc. Utiliza-se de dados ou de categorias teóricas já trabalhados por outros pesquisadores e devidamente registrados. Os textos tornam-se fontes dos temas a serem pesquisados. O pesquisador trabalha a partir das contribuições dos autores dos estudos analíticos constantes dos textos.

Foram levantadas informações relativas ao tema do projeto, a fim de analisar a metodologia ágil *Extreme Programming*, estudando suas práticas, princípios e valores.

Já a pesquisa documental é compreendida por Severino (2007, p. 122 - 123) como:

No caso da pesquisa documental, tem-se como fonte documentos no sentido amplo, ou seja, não só de documentos impressos, mas sobretudo de outros tipos de documentos, tais como jornais, fotos, filmes, gravações, documentos legais. Nestes casos, os conteúdos dos textos ainda não tiveram nenhum tratamento analítico, são ainda matéria-prima, a partir da qual o pesquisador vai desenvolver sua investigação e análise.

Foram levantadas pesquisas em artigos e revistas eletrônicas com o objetivo de ampliar a pesquisa para desenvolver uma análise sobre esta metodologia.

Além disto, foi abordado um estudo de caso referente à implantação da *Extreme Programming*. De acordo com Severino (2007, p.121), um estudo de caso é uma “Pesquisa que se concentra no estudo de um caso particular, considerado representativo de um conjunto de casos análogos, por ele significativamente representativo. [...]”.

Os dados relatados foram levantados à partir de pesquisa bibliográfica em livros, sites, dentre outros, referentes a metodologia ágil. Com o intuito de demonstrar o resultado da implantação da *Extreme Programming* em projetos de desenvolvimento de sistemas.

2. METODOLOGIA ÁGIL E *EXTREME PROGRAMMING*

2.1. INTRODUÇÃO À METODOLOGIA ÁGIL

Metodologia de desenvolvimento de sistemas tem como princípio um conjunto de práticas que auxiliam o desenvolvimento de um sistema. O resultado do produto final se define em como todo o processo foi desenvolvido.

Há diversas metodologias para o desenvolvimento de sistemas, entretanto as duas metodologias mais utilizadas são:

- Metodologias tradicionais – Focam na documentação e o planejamento do projeto. São utilizadas em projetos complexos e críticos, onde é necessário uma análise detalhada nos requisitos.
- Metodologias ágeis – Enfatizam o desenvolvimento do sistema. São adequadas para projetos onde é frequentemente a mudança dos requisitos.

Para Koscianski e Soares (2007 *apud* SOMMERVILLE, 2003) existem algumas atividades essenciais em comum a todas as metodologias, elas são:

- Especificação: identificar e definir as funcionalidades e características do sistema.
- Projeto e implementação: de acordo com as definições da especificação o software é produzido. Nesta etapa são propostos os modelos através de diagramas e protótipos que serão implementados no código por meio de alguma linguagem de programação.
- Validação: são executados revisões e testes, para garantir que os requisitos sejam atendidos.
- Evolução: é feito a manutenção do produto, como adaptar o produto para novas necessidades do cliente.

Algumas organizações trabalham sem utilizar nenhuma metodologia. Isto ocorre pelo motivo de que processos tradicionais não se conciliam às necessidades da organização, seja por motivo de insuficiência de recursos ou insatisfação da metodologia. Assim o projeto se resulta em baixa qualidade, atraso na entrega, alteração dos custos e não atende as expectativas do cliente.

A metodologia ágil tem o objetivo de auxiliar projetos, onde o desenvolvimento do software sofre alterações constantes nos requisitos. De acordo com Koscianski (2007), “A metodologia ágil se consiste em aceitar a mudança em vez de tentar prever o futuro”.

Entre as décadas de 1980 e 1990, existia um conceito de que o melhor modelo para se obter um sistema de qualidade, era adotado por um planejamento de projeto bem analisado e um rígido desenvolvimento de *software*. Esses sistemas geralmente eram sistemas de grande e longa vida, um exemplo é um sistema de controle de um avião, que algumas vezes pode durar até 10 anos até está pronto por completo.

Entretanto, quando foi aplicado este modelo de desenvolvimento para pequenas e médias empresas, teve-se um grande desapontamento. Por esta razão uma equipe de desenvolvedores sugeriu novos métodos ágeis. Assim a equipe de desenvolvedores se focaria mais no sistema, em vez da documentação. A Metodologia ágil se tornou popular somente em 2001, quando se foi apresentado e compartilhado alguns métodos ágeis e práticas comuns. Isto resultou o estabelecimento da aliança ágil.

As características chave desta metodologia são (Koscianski, 2007):

- Pessoas e interações ao invés de ferramentas e processos;
- Sistema executável ao invés da documentação;
- Participação do cliente em vez de apenas a negociação dos contratos;
- Retorno rápido a mudanças sem cumprir planos.

A metodologia ágil não recusa a documentação e o planejamento do projeto, mas apenas demonstra que o sistema executável tem uma importância maior em relação à parte documental. Assim esta metodologia se aproxima mais de projetos que necessitam ter respostas rápidas e se adaptarem às mudanças.

Um exemplo da utilização desta metodologia é quando o usuário não consegue passar todos os requisitos necessários para o desenvolvimento do sistema, isto pode acontecer por ele não conseguir prever como será o sistema final. Concluindo-se com a metodologia ágil pode-se evitar muitas falhas, como acontece em pequenos e médios sistemas trabalhados com a metodologia tradicional.

O conceito de métodos ágeis para Sommerville (2007, p. 273) é:

Métodos ágeis são métodos interativos de desenvolvimento que enfocam a especificação, projeto e implementação de sistemas incrementais. Eles envolvem diretamente o cliente no processo de desenvolvimento. A redução de overhead de desenvolvimento pode fazer o desenvolvimento de software ser o mais rápido possível.

2.2. EXTREME PROGRAMMING

A *Extreme Programming* é uma das metodologias ágeis mais conhecida e utilizada em equipes pequenas e médias para o desenvolvimento de sistemas. É baseada em sistemas onde os requisitos são incertos e sofrem mudanças frequentes.

Esta metodologia foi desenvolvida por Kent Beck em março de 1996 durante seu trabalho em Daimler Chrysler. O nome “*Extreme*” foi abordado pelo fato de aplicar em nível extremo a interatividade no desenvolvimento e a participação contínua dos clientes. A finalidade da XP é tornar rápido o desenvolvimento do projeto, atender as necessidades do cliente e assim consequentemente cumprir as estimativas de tempo e custo.

A XP é formada por um conjunto de princípios e valores que adotam as melhores práticas sobre a Engenharia de Software. A aplicação destas práticas ágeis resulta em um agradável ambiente de desenvolvimento de sistemas para a equipe e clientes do projeto. Como o desenvolvimento do projeto é de uma maneira “ágil”, o intuito é aumentar o máximo de produtividade realizada em cada dia, para que o cliente tenha um alto retorno do que foi investido.

No Brasil a metodologia está tendo uma grande aceitação e adaptação no desenvolvimento de sistemas comparado com as metodologias tradicionais. Suas práticas estão sendo utilizadas em empresas e em ambientes acadêmicos com base na qualidade do software oferecida. Por ser uma metodologia recente, há algumas alterações para pequenos reajustes, com o objetivo de melhorar o desempenho nos projetos.

O sistema é desenvolvido por incrementos, através das descrições feitas pelo usuário, essas descrições são denominadas como história de usuários. Durante o desenvolvimento, o usuário poderá identificar novas necessidades para serem implementadas no sistema. De acordo com Sommerville (2007),

[...] os requisitos são expressos como cenários (chamados histórias do usuário), que são implementados diretamente como uma série de tarefas. Os programadores trabalham em pares e desenvolvem testes para cada tarefa antes da escrita do código. Todos os testes devem ser executados com sucesso quando um novo código é integrado ao sistema. Há um pequeno espaço de tempo entre os releases do sistema.

Na Figura 1 é demonstrado o processo para implantar um incremento no sistema que está em desenvolvimento.

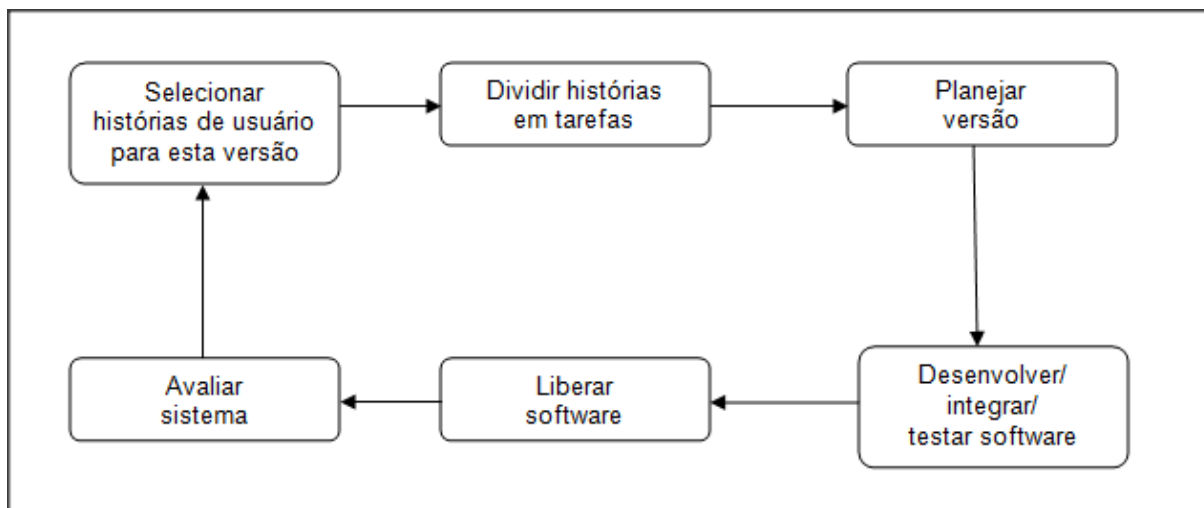


Figura 1 Ciclo de um *release* em *Extreme Programming*

Fonte: Adaptada de Sommerville, 2007, p.264.

2.2.1 VANTAGENS E RESTRIÇÕES

Através das práticas da XP, o projeto de desenvolvimento ganha uma grande vantagem de eficiência e eficácia no sistema. Essas vantagens são obtidas pelo fato de que as práticas são trabalhadas juntas e assim se tem um desempenho maior do que se cada uma fosse individualmente.

Os pontos mais fortes são:

- Com a participação do cliente no projeto, o resultado final do produto tende ser mais próximo que o cliente pediu;
- O planejamento na XP, ao contrário de outras metodologias, é feita no decorrer do projeto, sendo que é o cliente que decide o que vai ser feito primeiro e o que pode esperar. Isso torna o projeto passível de alterações a qualquer momento;
- Como a programação é feita em dupla, a ocorrência de erros é menor comparando-se com a programação individual;

- No início do projeto é apresentado um protótipo do sistema para o cliente, e este protótipo é utilizado para iniciar o desenvolvimento, deste modo é separada a parte da interface com a codificação;
- A entrega de várias versões com os ajustes de melhorias faz com que o cliente possa analisar se todas as suas necessidades foram atendidas e se não, ele recorre à equipe de desenvolvimento, assim a equipe não precisará fazer grandes modificações para corrigir o sistema;
- Com a utilização de um protótipo, a codificação se torna mais rápida pelo fato do desenvolvimento ser padronizado;
- A apresentação do protótipo ajuda o cliente a verificar o *layout* e as funcionalidades do sistema, desta maneira o cliente não precisa ficar aguardando muito tempo para visualizar o sistema, como ocorre nas metodologias tradicionais;
- A XP oferece qualidade no sistema, confiança aos clientes, cumpre as datas de entregas e atinge as metas de custo.
- Por ser flexível, simplifica nas atribuições das atividades.

Como toda metodologia tem suas restrições, com a XP não seria diferente, é necessário identificá-las para se adequar ao projeto.

Os pontos a serem trabalhados na XP são:

- A quantidade de membros da equipe é limitada, pois exige certo nível de maturidade da equipe;
- A documentação não é focada nas burocracias que as metodologias tradicionais exigem, como controles rígidos, avaliações de riscos, formulários, etc. Portanto, os sistemas que exigem esta parte burocrática não se adequam a XP;

- Como o método foca na interatividade com o cliente, fica complexo manter o tempo todo esta comunicação, tendo em vista que o cliente possui outras coisas a serem feitas. Para solucionar isso é necessário que alguns membros da equipe tenham o conhecimento necessário do negócio do cliente;
- Cada organização tem sua determinada cultura e o processo de mudança é difícil, deste modo, é necessário o comprometimento de todos para adotar as práticas da XP.

3. VALORES E PRINCÍPIOS BÁSICOS DA XP

A *Extreme Programming* adota um conjunto de valores e princípios que servem como guias para as práticas da XP. As práticas são ações que auxiliam o desenvolvimento do sistema. Os valores são a razão das atitudes tomadas no projeto. Os princípios fazem uma ponte entre os valores e as práticas da XP. Neste capítulo será detalhado os valores e os princípios da XP.

3.1. VALORES

É essencial a comunicação em um projeto, pois é através dela que se pode ter um melhor relacionamento entre os membros da equipe e entre a equipe com os clientes. Deste modo, a comunicação auxilia a explorar novas ideias, a transmitir essas ideias e então chegando à compreensão de todos.

A simplicidade tem como objetivo criar-se um código enxuto (com um menor número de métodos e classes) evitando funções desnecessárias, pois quanto mais simples for o software, será mais simples para o desenvolvimento, os testes e também a manutenção. De acordo com Ambler (2004, p.36):

“[...] A questão fundamental é manter seus modelos tão simples quanto possível, modelar hoje apenas para satisfazer às necessidades presentes e se preocupar posteriormente com as necessidades de modelagem futuras”

Portanto a implantação de um sistema simples será de uma forma rápida, e se houver modificações futuras o custo será menos do que se fosse desenvolvido um sistema complexo no início.

O *feedback* é a verificação de que os resultados estão corretos em relação ao sistema. Os programadores terão o retorno frequente do código e do cliente. Com relação ao código, é feito testes frequentes, que verificam os erros e o desempenho do sistema. E o cliente terá sempre um retorno da parte funcional do sistema para ele avaliar. Isto facilita para a identificação dos erros e para as correções de novas versões.

Para implantar esses três valores é essencial a coragem. É necessário que toda a equipe tenha espírito de coragem. Pois no decorrer do projeto é preciso coragem para inserir novas aplicações, modificar algo visando à máxima segurança ou até mesmo “jogar tudo no lixo” do que já foi produzido.

O alicerce destes valores é o respeito, é fundamental para qualquer projeto. Deve se ter o respeito entre a equipe, aceitando críticas e opiniões diferentes. Se não houver o respeito, o projeto dificilmente terá bons resultados.

3.2. PRINCÍPIOS BÁSICOS

Auto-semelhança

Este princípio recomenda que, se a equipe XP encontrar soluções para um determinado contexto, essas soluções podem ser adotadas em outros, mesmo que seja em escalas diferentes. Deste modo, é permitida a reutilização do código em outras situações no próprio sistema ou em outros. Entretanto os desenvolvedores devem ficar atentos para evitar uma replicação no código, ele deve identificar a necessidade da utilização do código e assim podendo reutilizá-lo. (TELES, 2006)

Benefício Mútuo

O benefício mútuo tem uma grande importância em projeto XP, pois através deste princípio todos no projeto são beneficiados de uma forma recíproca. Porém a adoção deste princípio é talvez uma das mais complicadas, visto que os projetos de desenvolvimento geralmente são complexos e acaba sofrendo pressão o que interfere na equipe, e desta maneira a equipe pode acabar beneficiando alguns, e prejudicando a outros. Logo, essas práticas que não são recíprocas a todos os membros da equipe, pode até romper o relacionamento entre eles e prejudicar o projeto. (TELES, 2006)

Diversidade

Para projetos de desenvolvimento de sistemas serem bem sucedidos requerem diversas habilidades, visando ter opiniões de diferentes profissionais com áreas distintas, deste modo à equipe terá ideias criativas e inovadoras para serem

implantadas no projeto. Entretanto, para isto é necessário que cada membro da equipe tenha pontos de vistas diferentes, pois se todos pensarem do mesmo modo, a equipe ficará restrita para explorar suas ideias. Opiniões diferentes auxiliam a encontrar e incrementar soluções para os problemas no projeto. (TELES, 2006)

Economia

Na economia, o valor do dinheiro sofre alterações ao longo do tempo. O valor do dinheiro hoje não será o mesmo do amanhã. Por este motivo, o sistema deve ser desenvolvido e implementado o mais rápido possível, para trazer o retorno do que foi investido. Deve ser controlado o tempo para que o dinheiro investido seja aplicado no projeto durante o desenvolvimento, sendo distribuído pelas etapas. Através do desenvolvimento iterativo, facilita para que cada semana novas funcionalidades sejam entregues, mesmo que o sistema ainda não esteja totalmente concluído.

Outro conceito importante na economia, é que através da XP é utilizadas técnicas como, por exemplo, manter o código enxuto e organizado, para tornar mais simples para futuras modificações. E com isso evitam-se gastos desnecessários no projeto. (TELES, 2006)

Falha

Estudos e experimentos são muito importantes em um projeto, pois assim a equipe sempre adquire conhecimentos novos. Quando um devido problema já tem uma solução, não é preciso pensar em alternativas e sim colocar em prática. Porém, quando não há soluções para algum problema, o melhor a se fazer é estudar e experimentar diferentes abordagens, para encontrar uma solução, deste modo o tempo de aprendizado será menor do que se houvessem reuniões e discussões para as respostas. Fazendo esses experimentos, mesmo que não haja sucesso em todos, pode ser uma grande fonte de aprendizado. Esta talvez possa ser uma das formas mais rápidas e importantes de aprender, que é explorar novos conhecimentos e praticá-los, mesmo que não de certo e tenha que voltar atrás e tentar outras alternativas. Com o *feedback* na XP, se tem o retorno concreto, então pode fazer tentativas, e não somente debates de possíveis alternativas. (TELES, 2006)

Fluidez

Em desenvolvimento de sistema há várias idas e vindas durante cada avanço. E quando as etapas são bem definidas faz com que essa liberdade de “ir e vir” sejam prejudicados. Na XP é proposto que ao invés de criar obstáculos, tendo que definir cada etapa, deve-se estudar o requisito para que a implementação do mesmo seja rápida.

As atividades são intercaladas até que a funcionalidade a ser implementada esteja finalizada. Neste processo será desenvolvido a análise do sistema, o design, testes e a implementação e essas etapas serão um ciclo até que sejam concluídas.

Com a participação do cliente nos processos, a equipe tem o *feedback* constante sobre as funcionalidades desenvolvidas e assim podem ser implementadas rapidamente para o usuário. Os clientes frequentemente recebem novas funcionalidades, o que ajuda a diminuir a ansiedade de espera e dando um retorno para a equipe, agilizando o projeto. (TELES, 2006)

Humanismo

A programação é uma atividade complexa que exige grande concentração. Pois envolve diversas técnicas, experimentos, criatividade, entre outras habilidades. Entretanto, infelizmente algumas propostas para a melhoria do desenvolvimento de sistema não respeitam a natureza humana. Como por exemplo, as que propõem que os desenvolvedores escrevam documentos complexos, antes de implementar as funcionalidades.

Alguns fatores que interferem na qualidade do trabalho do desenvolvedor são:

- **Segurança básica**- São os fatores físicos ou psíquicos, por exemplo, não estar com dores, não estar com fome, não estar se sentindo ameaçado e diversos fatores que não trazem a segurança.

- **Realização**- Contribuir a sociedade, através de suas habilidades e capacidade.

- Participação- Se identificar com o grupo, cumprir as responsabilidades adquiridas e passar confiança a equipe para alcançar objetivos comuns a todos envolvidos.

- Crescimento- Adquirir mais conhecimento, oportunidade de aperfeiçoar as habilidades.

- Intimidade- Ter intimidade com o grupo, podendo expor suas opiniões, sugerir ideias, participando de todas as etapas.

- Orgulho- Desenvolver trabalhos de qualidade para que possa sentir admiração.

Com as práticas da XP, é exploradas habilidades de cada um, aproveitando o máximo o melhor que podem oferecer, desenvolvendo um trabalho de qualidade. (TELES, 2006)

Melhoria

O software deve ser aperfeiçoado ao longo do tempo, atualizando novas funcionalidades necessárias, aprimorando o design, melhorando o desempenho, entre outros aspectos. As melhorias e manutenções no projeto devem ser contínuas. Pois se não houver esse cuidado, o software perderá seu devido valor.

Estudando melhor o sistema, maior será a chance de melhorá-lo, e também é uma maneira de adquirir novos conhecimentos. Como exemplo da utilização, se tem os ciclos semanais que permitem implantar o conhecimento adquirido durante a semana. (TELES, 2006)

Oportunidade

Em todos os projetos acontecimentos inesperados podem ocorrer, porém o modo em que a equipe reage diferencia o problema como um obstáculo ou uma oportunidade. Quando isto ocorrer, o melhor a se fazer é aprender o máximo, tendo

a oportunidade de encontrar novas soluções e melhorias para se implantar no projeto.

Na XP esta oportunidade de aprendizado é muito bem adotada, pois é uma experiência proveitosa para toda a equipe. (TELES, 2006)

“Passos de bebê”

Ao criar inúmeras linhas de código sem compilar por etapas, o resultado são mensagens de erro no sistema, e isso acaba causando:

- *Stress*;
- Preocupações;
- Frustração;
- Ansiedade;
- Desânimo;
- Desmotivação.

Este princípio determina que as etapas devam ser avançadas um pouquinho de cada vez, tratando os erros detalhadamente um por um. E assim evitando o aborrecimento de ter trabalhado duro no código e ter que refazê-lo.

O princípio não é adotado apenas na programação, ele está envolvido em todo o projeto. Se os erros são descobertos rapidamente, será mais fácil para encontrar soluções. (TELES, 2006)

Qualidade

Na *Extreme Programming* os valores são retornados rapidamente para se evitar desperdícios.

Se o sistema resultar em baixa qualidade terá grandes perdas, resultando:

- Prejuízo para o negócio;
- Descontentamento do cliente;
- Conflito entre o cliente e a equipe;
- Ansiedade;
- Perda de tempo para corrigir os erros;
- Desgaste da equipe;
- Perda de confiança;
- Dificuldade na adaptação do sistema de maneira segura;
- Insegurança;

A equipe XP pressiona para a criação de um sistema de alta qualidade. Criando um software de altíssima qualidade, torna o software mais satisfatório e econômico.

Infelizmente a qualidade é um dos itens que é mais sacrificado no projeto. Em algumas condições a equipe tem que tomar essa atitude para avançar algumas etapas com mais rapidez. (TELES, 2006)

Reflexão

A equipe XP, além de desenvolver seus trabalhos, analisam o modo que estão trabalhando e o objetivo do trabalho. Também refletem o porquê dos sucessos e dos fracassos obtidos. Não escondem seus erros e sim os expõe para trabalhar e aprender com eles.

Deve haver um equilíbrio entre a reflexão e o trabalho, a reflexão não pode durar tempo demais. Em alguns desenvolvimentos, a equipe se mantém tão focada pensando no desenvolvimento do sistema que acabam não tendo tempo para desenvolver. A reflexão deve ser feita depois da ação. O aprendizado é o resultado da ação refletida. Na XP para intensificar o *feedback*, as reflexões da equipe são intercaladas com a ação. (TELES, 2006)

Redundância

Para a solução de um problema, deve ser resolvido de diversas formas diferentes. Se uma solução falhar, as outras darão o apoio para poder prevenir de uma crise no desenvolvimento.

Os defeitos são muitos críticos e difíceis de resolver. A XP trata os defeitos através de diversas práticas. Mesmo que algumas dessas práticas sejam redundantes, tratando dos mesmos defeitos, servem para propósitos válidos. (TELES, 2006)

Responsabilidade aceita

A responsabilidade não é atribuída e sim aceita. Se uma responsabilidade é passada para alguém, somente a pessoa decidirá se irá aceitar ou não. (TELES, 2006)

4. PRÁTICAS DA XP

4.1. CLIENTE DISPONÍVEL/PRESENTE

Na XP durante todo o projeto de desenvolvimento do sistema, é essencial a presença e participação do cliente. Ele deve apoiar a equipe, estando sempre disponível para esclarecer as dúvidas dos requisitos, que poderão ser alterados.

O cliente é considerado como um integrante da equipe, sua função é trazer os requisitos para o desenvolvimento do sistema e validá-los. (KOSCIANSKI, 2007)

4.2. JOGO DE PLANEJAMENTO

De acordo com Koscianski e Soares (2007, p.196) planejamento “se consiste em decidir o que é necessário fazer e o que pode ser adiado no projeto”. O planejamento é feito em reuniões semanais entre o cliente e a equipe. Através das reuniões os requisitos são escritos em cartões de histórias pelo cliente, esses cartões são implementados para a criação do escopo. A figura 2 é um exemplo de um cartão de história, deve ser especificada a história sem muitos termos técnicos.

Data: _____	Tipo de história: __ Nova __ Melhoria __ Debug		
Número: _____	Prioridade: _____	Risco: _____	Estimativa: _____
Descrição:			
Notas:			
Acompanhamento:			
Data	Status	A fazer	Comentário

Figura 2 Cartão de história

Fonte: Adaptada de Koscianski e Soares , 2007, p.199.

A história pode ser dividida em tarefas. A tarefa deve ser descrita no cartão de tarefa como na figura 3.

Data: _____	Número: _____	Estimativa: _____	
Responsável 1: _____		Responsável 2: _____	
Descrição:			
Notas:			
Acompanhamento:			
Data	Status	A fazer	Comentário

Figura 3 Cartão de tarefa

Fonte: Adaptada de Koscianski e Soares , 2007, p.199.

O acompanhamento deve ser em curtos intervalos de tempo. Com o fato de o cliente estar presente, há grande aproveitamento nas reuniões.

4.3. *STAND UP MEETING*

São reuniões curtas, tendo no máximo o limite de 10 minutos. A reunião é feita com todos os envolvidos em pé, para tornar a reunião rápida e objetiva. Devem ser praticadas todos os dias, assim a equipe acompanha os avanços e se mantém informados sobre o andamento do projeto. (KOSCIANSKI, 2007)

4.4. *PROGRAMAÇÃO EM PAR/DUPLA*

O desenvolvimento do sistema é feito em dupla. O mais inexperiente digita o código e o outro observa o código que está sendo trabalhado, identificando se há erros e pensando em como solucioná-los. (KOSCIANSKI, 2007)

4.5. REFACTORING (REFATORAÇÃO)

A refatoração é a reutilização do código-fonte. O desenvolvedor tem como base o código já pronto, assim ele faz melhorias para torná-lo mais confiável.

Reutilizando o mesmo código, a leitura se torna mais simples e fácil. Esta prática simplifica o código e auxilia para futuras implementações no sistema. (KOSCIANSKI, 2007)

4.6. DESENVOLVIMENTO GUIADO POR TESTES

A realização de testes é muito importante para o desenvolvimento do software. Quanto antes forem executados, mais rápido será para verificar se há erros e assim encontrar soluções rapidamente.

A maioria das metodologias não dá tanta importância para os testes, porém para a XP os testes são de alta importância. (TELES, 2006)

4.7. CÓDIGO COLETIVO

Todos os integrantes da equipe podem ter o acesso ao código, possibilitando a refatoração e a liberdade de fazer alterações necessárias no código. Assim todos podem participar do desenvolvimento. (KOSCIANSKI, 2007)

4.8. PADRÕES DE DESENVOLVIMENTO

Recomenda-se a padronização no desenvolvimento do código. Todos os integrantes da equipe devem adotar os padrões para facilitar o entendimento do código, favorecendo uma aprendizagem para a equipe. (KOSCIANSKI, 2007)

4.9. DESIGN SIMPLES

O sistema deve atender os requisitos atuais, sem ter a preocupação com os futuros. O código sendo simples e padronizado facilita a compreensão de todos da equipe e facilita também para futuras funcionalidades a serem implementadas. (KOSCIANSKI, 2007)

4.10. METÁFORA

Para facilitar a comunicação entre a equipe e o cliente eles devem falar a mesma língua, sem utilizar termos técnicos, pois em cada área se tem os próprios termos técnicos, e se não entrarem em uma compreensão de ambos, poderá ter ambiguidade na comunicação e resultando em falhas no projeto. (KOSCIANSKI, 2007)

4.11. RITMO SUSTENTÁVEL

Para um bom rendimento no trabalho, a XP recomenda que a jornada de trabalho seja 8h diárias, sendo o total de 40h semanais. Isto garante que o ambiente de trabalho seja saudável. O projeto terá um maior rendimento se a equipe estiver concentrada. (TELES, 2006)

4.12. INTEGRAÇÃO CONTÍNUA

A XP sugere que todas as etapas do projeto, devem ser interligadas e serem realizados testes frequentes durante o dia, com isto a equipe obtém uma integração contínua do projeto. Diferente se as etapas fossem divididas e trabalhadas individualmente por cada responsável. (TELES, 2006)

4.13. RELEASES CURTOS

Consiste em entregas frequentes de pequenas funcionalidades do sistema, assim o cliente visualiza algumas partes do sistema funcionando, e podendo analisar se está atendendo suas necessidades. (TELES, 2006)

5. EQUIPE XP

De acordo com Teles (2006), o objetivo da equipe é que cada um contribua com o seu melhor para o projeto, com o intuito de que a equipe tenha sucesso. Os papéis não são fixos, o importante é que cada um contribua com o seu melhor. Os papéis empenhados são descritos neste capítulo.

5.1. GERENTE DE PROJETO

O gerente do projeto faz a comunicação entre os clientes, a equipe e os eventuais fornecedores. Ele é responsável pela administração do projeto.

Os gerentes de projeto devem auxiliar a equipe para que a mesma possa lidar com os fatores e pressões externas. Deve motivar a equipe e ter uma boa relação com o cliente, para que ele possa ter uma maior participação no desenvolvimento do projeto. Para que sua equipe tenha um bom desempenho no trabalho é preciso que ele siga os valores, princípios e práticas do XP.

5.2. COACH

A responsabilidade do *coach* (técnico) é trabalhar com as questões técnicas do projeto, como o comportamento da equipe no projeto e assim sinalizar os possíveis erros realizados. Ele deve ter um amplo conhecimento na área de desenvolvimento do projeto.

5.3. ANALISTA DE TESTE

O analista de teste deve ter uma visão parecida com o cliente, então se sugere que o analista não tenha conhecimento em desenvolvimento, assim evita-se uma visão tendenciosa. Esta pessoa é responsável pela qualidade do sistema, onde é feito os testes escritos. Ele auxilia o cliente a descrever os casos de testes e verifica se o sistema está atendendo todos os casos de testes.

5.4. REDATOR TÉCNICO

A função do redator técnico é documentar os sistemas, tendo uma maior dedicação no desenvolvimento da codificação. A documentação deve descrever o código escrito e as regras de negócios que foram atendidas pelo software.

5.5. DESENVOLVEDOR

O desenvolvedor é responsável pela análise, projeção e codificação do sistema. Também é responsável por desenvolver os testes e as funcionalidades para o sistema. Há vários níveis de desenvolvedores dentro de uma equipe, mas através das práticas do XP, a tendência é que a equipe padronize seus trabalhos.

6. CICLO DE VIDA DO PROJETO XP

Para Costa (2011), um projeto XP é dividido por algumas fases ao longo do desenvolvimento. Essas fases são tarefas que são executadas pela equipe. Este capítulo traz uma explicação das fases de um projeto XP, objetivando demonstrar a fluidez do projeto.

6.1. FASE DE EXPLORAÇÃO

Esta fase é desenvolvida antes da elaboração do sistema. É investigado as possíveis decisões e soluções de problemas do sistema. Os programadores analisam como o sistema irá interagir com o ambiente tecnológico (como exemplo, rede, software, hardware, performance, etc.) onde o sistema vai atuar. Nesta etapa o cliente e os programadores começam a criar confiança. É sugerido o tempo máximo de duas semanas para esta fase, após esta etapa já pode ser construído o primeiro *release* (as funcionalidades bem definidas) do sistema. (Costa, 2011)

6.2. FASE DE PLANEJAMENTO INICIAL

É definida a data de divulgação do primeiro *release*, e também para a definição do escopo do sistema. Nesta fase os programadores junto com o cliente trabalham com as histórias (utilizada para requisitos do sistema, é escrita na linguagem simples) a serem implementadas, onde os programadores definem quais delas poderão ser implantadas, isso irá depender da velocidade de implementação.

Definido a quantidade de histórias, é feito o planejamento de iteração (contém um conjunto de histórias a serem implementadas), para que o cliente defina quais as histórias de maior necessidade serão implementadas.

Para a etapa de iteração é gasto de uma a três semanas, já para o *release* o tempo gasto varia de dois até quatro meses. (Costa, 2011)

6.3. FASE DE ITERAÇÕES DO *RELEASE*

Nesta etapa são elaborados os casos de testes das funcionalidades do sistema. Os programadores seguem um fluxo de atividade para cada iteração:

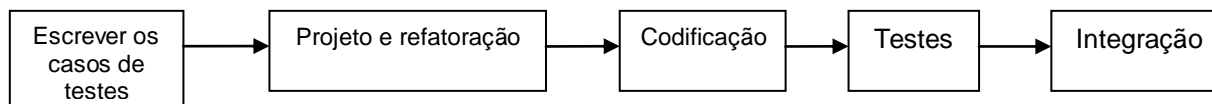


Figura 4 Fluxo de atividades

Fonte: Adaptada de Costa (2011)

Seguindo esse fluxo, o sistema vai sendo montado e ao finalizar o primeiro *release*, já se tem um resultado para domínio do problema. Assim, os próximos *releases* poderão ser mais curtos e assim tendo estimativas mais confiáveis.

6.4. FASE DE PRODUÇÃO

Cada *release* do sistema depois de finalizado é testado em um ambiente onde simula o ambiente real que irá ser aplicado. Deste modo é analisado como o sistema irá atuar no seu ambiente de produção, visualizando o desempenho e comportamento do mesmo. (Costa, 2011)

6.5. FASE DE MANUTENÇÃO

Com o sistema já existente e rodando, podem-se implantar novas funcionalidades, melhorar o código, utilizar novas tecnologias, incrementar novos mecanismos, entre outras melhorias.

Porém para a aceitação de novos mecanismos deve se ter cuidado, pois a utilização de novos recursos pode paralisar o sistema, assim trazendo prejuízo para o cliente. Antes de alguma mudança deve ser bem estudado antes. (Costa, 2011)

6.6. FASE DE ENCERRAMENTO DO PROJETO

É a finalização do projeto XP. Há dois tipos para o término do projeto, um é quando o sistema atendeu todas as necessidades do cliente e não precisa mais implementar novas funcionalidades e o outro é quando o custo do projeto se torna inviável. (Costa, 2011)

7. ESTUDO DE CASO: IMPLANTAÇÃO DA *EXTREME PROGRAMMING* NA EMPRESA ANCAR

O presente estudo de caso foi baseado no artigo da empresa Ancar, está disponível no site da Improve IT, tal estudo foi desenvolvido por Vinícius Teles da Improve IT junto da equipe da Ancar.

A Ancar é uma empresa nacional, localizada no Rio de Janeiro, atua em administração de shopping centers há mais de 75 anos. A partir da década de 80 a Ancar iniciou a informatização na empresa, eram utilizados softwares de prateleira para os serviços de Tecnologia da informação, porém houve a necessidade de implantar um sistema de faturamento interno, o sistema foi desenvolvido por apenas um único desenvolvedor, o sistema era em Clipper. Este software foi muito utilizado durante as décadas de 80 e 90.

Ao se iniciar o ano 2000, com a mudança no mercado surgiu novas necessidade para desenvolver um novo sistema. O sistema implantado foi o SGSC (Sistema de Gestão de Shopping Centers), desenvolvido em Delphi com a base dos dados em MS SQL Server. Este sistema foi utilizado por vários shoppings que são administrados pela Ancar.

No ano de 2004 a empresa se interessou pela metodologia ágil *Extreme Programming* através da participação de sua equipe em uma reunião do Grupo XP Rio e assim contratando os serviços da Improve IT. O treinamento teve início no primeiro semestre de 2005, a equipe iniciou com um treinamento sobre orientação a objetos utilizando a linguagem Java e a metodologia XP. O treinamento foi em alguns dias por semana, para que a equipe pudesse conciliar o treinamento com suas atividades diárias. No segundo semestre do ano de 2005, a Ancar se mudou para um escritório com um espaço de trabalho mais aberto, e iniciou o desenvolvimento de um sistema com a utilização da metodologia *Extreme Programming*.

Com o sistema SGSC ocorriam falhas na acessibilidade, pois não fornecia as informações de maneira apropriada, não atendia os níveis estratégicos da empresa. Por este critério a Ancar investiu para o desenvolvimento de um novo sistema

chamado Ancar2Go, para que pudesse agregar todas as informações estratégicas da empresa, facilitando a acessibilidade para os diretores, assim eles podiam estar conectados na rede interna ou externa.

O desenvolvimento do sistema teve início em agosto de 2005, foi elaborado pela equipe de seis pessoas da Ancar. O sistema foi desenvolvido com a linguagem Java com interface desktop com o acesso através do Java Web Start. Em todas as etapas foram utilizadas as práticas da XP. As iterações do sistema foram semanais e no jogo do planejamento o cliente escrevia as histórias do sistema utilizando os cartões de histórias em papel. Os desenvolvedores estipularam o esforço em pontos, em cada ponto eram estimado 4 horas de trabalhos adequadas.

Todos os dias a equipe realiza um *stand up meeting* no período da manhã, eles revisavam todo o trabalho do dia anterior, estipulavam as atividades de prioridade para o dia e atualizavam o andamento diário do projeto. A mesma equipe que foi responsável pelo desenvolvimento do sistema Anca2Go foi também responsável pela a manutenção do SGSC. Após cada iteração concluída a equipe avaliava como foi o desempenho da semana, quais os pontos que deviam ser melhorados para as próximas iterações. Para a avaliação a equipe utilizaram ferramentas simples, como quadro branco, papeis, canetas e *post its*.

Ao longo do desenvolvimento, os programadores desenvolveram em pares, revezando frequentemente. A equipe trabalhou junta, a responsabilidade de cada tarefa era atribuída à equipe em geral.

Os aspectos positivos do projeto de acordo com Teles (2006) foram:

- **Integração Contínua**

Uma das práticas da XP de maior destaque no projeto foi a integração contínua. Para sinalizar a integração eles utilizaram um boneco de borracha do Bob Esboja, assim quando um par fosse integrar seu trabalho eles tomavam posse do boneco. Quando o boneco era apertado duas vezes o som produzido indicava que o projeto estava sendo iniciado. E quando ele era apertado uma única vez indicava

que era o término do procedimento. Com esta divertida forma de trabalho, criou-se a ligação entre as etapas.

- **Deployment Automatizado**

A equipe implantou a automação do *deployment* no projeto, permitindo produzir novas versões rapidamente.

- **Testes**

Houve dificuldades em testar o código, pois os testes exigem um maior conhecimento em orientação a objetos para a compreensão do resultado, então como apoio teve a participação do mentor da Improve IT, auxiliando a equipe uma vez por semana.

- **Programação Em Par**

A técnica da programação em par foi muito bem adotada desde o início, e cada vez mais se afirmou no projeto, trazendo bons retornos. Aumentou o nível de concentração dos desenvolvedores, criou confiança no código, aumentou a produtividade, o que trouxe grande satisfação. O gerente da equipe da Ancar apoiou esta prática desde o início, pois percebeu rapidamente os benefícios que a programação em par proporcionava para o projeto. Na Ancar esta prática parece ser muito bem alinhada aos valores da organização.

Em toda empresa se segue um conjunto de valores, no caso da Ancar, o que é mais valorizado é os relacionamentos humanos, onde haja respeito entre as pessoas para que o ambiente de trabalho seja saudável e agradável.

Esta técnica foi bem enquadrada nestes valores, pois os programadores trabalham de uma maneira descontraída e desenvolvem em um curto tempo. Uma característica interessante da programação em par é a fonte de conhecimento que é oferecida. Um exemplo citado por Teles (2006) foi,

[...] a Ancar contratou um novo desenvolvedor, chamado Celestino, com conhecimento e experiência em Delphi e MS SQL Server, mas pouco conhecimento técnico de Java. Programando em par, ele aprendeu a maior parte dos conceitos de Java com seus colegas em poucas semanas.

Uma vantagem que é oferecida pela XP é que caso algum desenvolvedor se ausente do projeto, isto não prejudicará o projeto.

- **Entrosamento da Equipe**

Um dos pontos marcantes do projeto foi o entrosamento da equipe, criando uma convivência divertida entre os membros. Desta maneira todos da empresa trabalham juntos.

Uma atividade adquirida para criar o entrosamento entre a equipe foi o malabarismo feito com bolinhas durante os intervalos, deste modo alguns desenvolvedores demonstravam suas habilidades, que ao invés de serem criticados por realizar essas atividades no trabalho, eram admirados pela coordenação e controle que apresentavam.

Outra atividade realizada foram os jogos de boliche realizados durante o almoço.

Através dessas ações interligadas a programação em par, auxiliaram a motivação da equipe e criaram um bom relacionamento entre eles.

Desafios Técnicos

O sistema tinha como objetivo trazer as informações críticas do negócio que deveria ser online e com um design simples, para que os usuários encontrassem os dados de uma forma rápida e acessível. Como deve se ter o controle das unidades comerciais dos shoppings, a equipe decidiu implementar as plantas do shoppings no sistema, desta forma era possível verificar quais estavam alugadas, as que estão desocupadas, entre outros fatores.

Entretanto, a implantação da planta no sistema gerou alguns desafios técnicos para a empresa. Como por exemplo, desenhar a planta, implementá-la no sistema, entre outros. Porém com as práticas da XP, a equipe buscou soluções gradativamente, estudando cada solução como é proposto na XP.

Relacionamento de TI com a organização

A utilização da Extreme Programming no projeto Ancar2Go, gerou uma grande aceitação para o restante da empresa. Isto se deve pelo empenho dos desenvolvedores em envolver os usuários no projeto e também envolver as diversas áreas da organização nos processos.

De acordo com a equipe Ancar no começo do projeto, as definições das funcionalidades foram atribuídas somente pelo diretor do setor de TI na empresa. Como ele desenvolveu o software em Clipper, ele conhecia as principais necessidades da empresa.

Contudo, com o andamento do projeto a equipe resolveu apresentar o sistema para os outros departamentos da empresa, o que resultou em um grande *feedback* dos outros setores. E desta forma, todos podiam participar nos processos.

Precisa melhorar

A XP gerou grandes benefícios para a Ancar, porém, há algumas melhorias para desenvolver. Os testes devem ter uma boa cobertura no sistema, para que o mesmo se mantenha sempre em bom estado.

Para trazer esta melhoria, deve se ter mais disciplina da equipe e evitar a ansiedade de desenvolver e já implementar a funcionalidade, e acabam pulando a execução de testes.

Extreme Programming no SGSC

No início o objetivo da Ancar era utilizar a Metodologia XP apenas para desenvolver o Ancar2Go, porém este trabalho motivou a Ancar para a manutenção do SGSC.

Através da XP, a equipe iniciou o planejamento no SGSC e realizaram as iterações semanais. Foi monitorado cada atividade de manutenção do sistema. E também foram criados alguns scripts para automatizar alguns processos no SGSC.

Para Teles (2006) “a equipe ainda tem pegar mais experiência, ainda se tem pontos que devem ser melhorados, mas a cada dia a equipe se torna cada vez mais habilidosas com o uso da XP”.

8. CONSIDERAÇÕES FINAIS

A utilização de uma metodologia inadequada em um projeto de desenvolvimento de sistemas pode resultar em diversas falhas, como atraso no cronograma, alteração dos custos, insatisfação do cliente, entre outros. O desenvolvimento pode começar bem, atingindo todos os objetivos, porém no decorrer do projeto, as falhas trazem grandes prejuízos para o projeto, onde desmotiva a equipe por ter trabalhado duro e não ter alcançado o objetivo final e cria a falta de confiança do cliente pela insatisfação do produto final, que na maioria das vezes não atende todos os requisitos necessários. Isto acontece por diversos fatores como: metodologias que são focadas mais na documentação do projeto, deixando o desenvolvimento como segunda prioridade, ou falta de comunicação no projeto, ou quando o cliente não consegue passar os requisitos necessários para a equipe e também quando não se tem a participação do cliente no projeto, e outros fatores que prejudicam o projeto.

A metodologia ágil *Extreme Programming* foi proposta para solucionar esses problemas utilizando soluções extremas. Esta metodologia utiliza as melhores práticas da Engenharia de software. O objetivo da *Extreme Programmig* é oferecer um sistema de qualidade que atenda todas às necessidades do cliente, dentro do prazo e custo definido. Esta metodologia é adequada para projetos onde os requisitos são vagos e mudam com frequência.

Ela busca o máximo de desempenho da equipe a cada dia de trabalho, assim em um curto tempo o cliente pode obter um sistema que possa ser utilizado, e deste modo podendo estudar e aprender com o mesmo, verificando se o sistema é realmente o desejado e logo trazendo um retorno para a equipe.

A XP segue algumas práticas, princípios e valores que visam nivelar a metodologia, fazendo que qualquer equipe possa adotar esta metodologia para desenvolvimento.

Para demonstrar os resultados das práticas da metodologia *Extreme Programming*, foram consultados alguns artigos que descrevem empresas que adotaram a metodologia. Os resultados desses estudos de caso foram de grande aceitação e satisfação. Um dos principais foi implementado pela empresa Improve It na organização Ancar que está detalhado neste trabalho. Através deste estudo de caso é possível visualizar os resultados positivos que esta metodologia oferece. O projeto foi desenvolvido em um ambiente saudável de trabalho, com a equipe unida, tendo o resultado final do projeto um sucesso.

Além de um ambiente mais favorável para o desenvolvimento do sistema, trouxe como benefícios: a interação e troca de conhecimentos entre todos os membros da equipe, resolução de problemas de uma forma mais ágil e eficiente, interpretações de formas diferentes em relação ao sistema proposto, resultado este que pode ser visto na conclusão do projeto.

Devido à adaptação para a nova metodologia no projeto, não foi possível estabelecer o resultado da adoção da *Extreme Programming* em relação ao custo e prazo do projeto.

Sendo assim, deseja-se que em trabalhos futuros possam obter mais dados em particular, ao prazo e custo para que se possa comparar com este estudo de caso.

Outro ponto a ser estabelecido em trabalhos futuros é o detalhamento dos testes funcionais, pois como foi descrito no estudo de caso a equipe não adotou corretamente a execução dos testes, quando era finalizada a funcionalidade já era implementada, pulando os testes. O que pode trazer prejuízos nos projetos futuros, como:

- Perda de tempo: A ineficiência na quantidade de testes pode trazer resultados em que o trabalho deve ser refeito e novos testes realizados, tomando mais tempo e tornando inviável o resultado.

- Aumento no custo: Pelo fato de acrescentar horas trabalhadas no projeto, é inevitável que seja alterado o custo para o desenvolvimento do sistema.
- Insatisfação do cliente: Com o aumento de tempo e custo no projeto, o vínculo de confiança do cliente com a equipe pode se desgastar ao longo do projeto.
- Desmotivação da equipe: O empenho e a dedicação da equipe no desenvolvimento são prejudicados, trazendo desmotivação ao ter que refazer o trabalho. Isto pode causar insegurança para a implementação das funcionalidades no sistema.

Segundo constatado no estudo de caso, houve deficiência de alguns membros da equipe para a realização dos testes por falta de conhecimento em orientação a objetos. Projetos futuros devem qualificar e realizar treinamentos nesta área para que venha suprir esta falta de conhecimento e habilidade nos testes.

A XP ainda está em crescimento e, como toda metodologia recente, ainda está realizando melhorias em alguns pontos que são restritos para alguns projetos. Cada vez mais as empresas estão adotando a XP, e a mesma vem se afirmando como uma das metodologias mais bem adotadas.

9. REFERÊNCIAS

AMBLER, S. W. Modelagem Ágil: Práticas eficazes para a Programação Extrema e o Processo Unificado. Tradução: Acauan Fernandes. Porto Alegre: Bookman, 2004.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. Citação: NBR-10520/ago - 2002. Rio de Janeiro: ABNT, 2002.

CAMORA, A.A. Desenvolvendo Sistemas com Extreme Programming. Disponível em:

<[http://www.fatecriopreto.edu.br/Storage/Projetos%20de%20Graduacao/Informatica%20para%20Negocios/Turma%2001%20\(12-2006\)/Desenvolvimento%20de%20sistemas%20Extreme%20Programing.pdf](http://www.fatecriopreto.edu.br/Storage/Projetos%20de%20Graduacao/Informatica%20para%20Negocios/Turma%2001%20(12-2006)/Desenvolvimento%20de%20sistemas%20Extreme%20Programing.pdf)>.

Acesso em: 13. Ago. 2013. 15h57.

CASADEI, C. et al. Metodologias Ágeis: Um Exemplo de Aplicação da Extreme Programming (XP) Disponível em:

<<http://www.fatecsaocaetano.edu.br/fascitech/index.php/fascitech/article/view/34/33>>

. Acesso em: 12 fev. 2013. 12h45.

CASTRO, V. A. Software e Empreendedorismo – Valores, Princípios e Práticas.

Disponível em: <<http://devagil.wordpress.com/2007/02/04/425-xp-respeito/>>. Acesso em: 29. Set. 2013. 17h40.

COSTA, G. H. de C. Engenharia de Requisitos no Desenvolvimento de Software Ágil. Disponível em: <

http://www.cin.ufpe.br/~in1020/arquivos/monografias/2010_2/Monografia_Gustavo_Carvalho.pdf>. Acesso em: 07.nov.2013. 10h32.

GUIMARÃES, C. P. Metodologias Ágeis Extreme Programming. Disponível em:

<<https://intranet.dcc.ufba.br/pastas/pub/MATA62t01/atividades/aula-12/03%20MA-XP.pdf>>. Acesso em: 16. out. 2013. 10h15.

INSTITUTO DE ENSINO SUPERIOR DE BAURU (IESB). Manual de trabalho de conclusão de curso (TCC). (2005). Disponível em: <http://www.iesbpreve.com.br/diversos/Manual_TCC_IESB.pdf>. Bauru/SP. Acesso em: 27 set. 2012. 20h00.

Koscianski, A; Soares M. dos S. Qualidade de software: aprenda as metodologias e técnicas mais modernas para o desenvolvimento de software. 2ª ed. São Paulo: Novatec Editora, 2007.

NAKASHIMA, E. Metodologias ágeis X Metodologias tradicionais. Disponível em <<http://tisimples.wordpress.com/2009/04/18/metodologias-ageis-x-metodologias-tradicionais/>>. Acesso em: 26 ago. 2013. 12h19.

PETRI, D. O que é Extreme Programming Parte 2. Disponível em: <<http://www.almbrasil.com.br/o-que-e-extreme-programming-parte-2/>>. Acesso em: 13. ago. 2013. 16h34.

PETRI, D. O que é Extreme Programming Parte 3. Disponível em: <<http://www.almbrasil.com.br/o-que-e-extreme-programming-parte-3/>>. Acesso em: 24 jan. 2013. 19h22.

PRESSMAN, R. S. Engenharia de Software. Rio de Janeiro: McGraw-Hill, 2002.

PRESSMAN, R. S. Engenharia de Software. Disponível em: <http://books.google.com.br/books?hl=pt-BR&lr=&id=y0rH9wuXe68C&oi=fnd&pg=PR5&dq=sommerville+engenharia+de+soft+oft&ots=AxPnuOzbO_&sig=P95xesd41Hdd7HcHeoVndE4-igY#v=onepage&q&f=false>. Acesso em: 02. set. 2013. 19h29.

SILVA, V. B. Práticas da Extreme Programming na Redução da Sobrecarga de Trabalho. Disponível em: <<http://fateczl.edu.br/TCC/2009-2/tcc-68.pdf>>. Acesso em: 15. ago. 2013. 16h20.

SILVÉRIO, K. M. Extreme Programming. Disponível em: <http://artigos.netsaber.com.br/resumo_artigo_12915/artigo_sobre_extreme_programming>. Acesso em: 05. nov. 2013. 13h34.

SOMMERVILLE, I. Engenharia de Software. 8ª ed. Tradução: Selma Shin Shimizu Melnikoff, Reginaldo Arakaki, Edílson de Andrade Barbosa. Revisão técnica: Keichi Kirama. São Paulo: Pearson Addison Wesley, 2007.

SOUZA, L. M. Método Ágil XP (Extreme Programming). Disponível em: <http://intranet.fainam.edu.br/aceso_site/fia/academos/revista3/6.pdf>. Acesso em: 14. fev. 2013. 14h50.

TELES, V. M. Extreme Programming. São Paulo: Novatec, 2004.

TELES, V. M. Princípios do XP. Disponível em: <<http://desenvolvimentoagil.com.br/xp/principios/>>. Acesso em: 20. out. 2013. 21h40.

TELES, V. M. Qualidade. Disponível em: <<http://desenvolvimentoagil.com.br/xp/principios/qualidade>>. Acesso em: 30. out. 2013. 13h10.

TELES, V. M. Valores do XP. Disponível em: <<http://desenvolvimentoagil.com.br/xp/valores/>>. Acesso em: 09. set. 2013. 18h10.

TELES, V.M. Estudo de caso Ancar. Disponível em: <<http://www.improveit.com.br/caseAncar.jsp>>. Acesso em: 10. nov. 2013. 09h34.

TELES, V.M. Entrevista com Edson de Lima na Série Experiências Ágeis. Disponível em: <<http://improveit.com.br/podcast/improvecast-22-entrevista-edson-de-lima-experienciais-ageis>>. Acesso: 11. nov. 2013. 12h30.

TELES, V. M. Extreme Programming. Disponível em: <<http://improveit.com.br/xp>>. Acesso em: 21. out. 2013. 09h37.

TELES, V. M. Um Estudo De Caso Da Adoção Das Práticas E Valores Do Extreme Programming. Disponível em: <<http://improveit.com.br/xp/dissertacaoXP.pdf>>. Acesso em: 16. out. 2012. 20h10.

WELLS, D. Extreme Programming: A gentle introduction. Disponível em: <<http://www.extremeprogramming.org/>>. Acesso em: 22 out. 2012. 15h30.