

MASSAYOHI STEVER CAMPOS PARDO

Integração das metodologias Extreme Programming, Scrum e PMBOK: estudo de caso sobre a viabilidade de suas adaptações e aplicabilidades

São Paulo
(2016)

MASSAYOHI STEVER CAMPOS PARDO

Integração das metodologias Extreme Programming, Scrum e PMBOK: estudo de caso sobre a viabilidade de suas adaptações e aplicabilidades

Monografia apresentada à Escola Politécnica da Universidade de São Paulo para obtenção do título de Especialista em Gestão e Engenharia da Qualidade.

Orientador: Professor Doutor Adherbal Caminada Netto.

São Paulo
(2016)

MASSAYOHI STEVER CAMPOS PARDO

Integração das metodologias Extreme Programming, Scrum e PMBOK: estudo de caso sobre a viabilidade de suas adaptações e aplicabilidades

Monografia apresentada à Escola Politécnica da Universidade de São Paulo para obtenção do título de Especialista em Gestão e Engenharia da Qualidade.

Orientador: Professor Doutor Adherbal Caminada Netto.

São Paulo
(2016)

MBAEQ
p214j

Catálogo-na-publicação

Pardo, Massayohi Stever Campos

Integração das metodologias Extreme Programming, Scrum e PMBOK:
estudo de caso sobre a viabilidade de suas adaptações e aplicabilidades / M. S.
C. Pardo – São Paulo, 2016.

76 p.

Monografia (MBA em Gestão e Engenharia da Qualidade) - Escola
Politécnica da Universidade de São Paulo. PECE – Programa de Educação
Continuada em Engenharia.

1.Metodologia 2.Processo 3.Gestão 4.Qualidade I.Universidade de São
Paulo. Escola Politécnica. PECE – Programa de Educação Continuada em
Engenharia II.t.

DEDALUS - Acervo - EPMN



31600023343

2772672

À minha mãe, Tereza Campos Pardo; aos meus irmãos, Alexandre Campos Pardo e Jhonne Campos Pardo; à minha filha, Miyuki Duque Campos Pardo e a toda minha família que não mediu esforços para o meu sucesso e evolução pessoal e profissional.

AGRADECIMENTOS

Agradeço, inicialmente, o Professor, Mestre e Doutor Adherbal Caminada Netto pela contribuição com o seu vasto conhecimento em suas orientações, sempre com sabedoria e maestria, tornando possível o desenvolvimento e a conclusão deste trabalho.

Ao Programa de Educação Continuada da Escola Politécnica da Universidade de São Paulo – PECE/Poli – pelo excelente curso e aos grandes Mestres e Doutores, que ministram suas aulas com brilhantismo, transmitindo o conhecimento necessário para que seus alunos aperfeiçoem suas virtudes, tornando-os ainda mais capacitados para o mercado de trabalho e vivências corporativas. Estes ilustres professores foram fundamentais para o meu desenvolvimento e aprendizado ao longo destes dois anos.

Agradeço à minha família, em especial, minha mãe Tereza Campos Pardo, por permanecer sempre ao meu lado em todos os momentos da minha vida.

Aos meus amigos, que souberam compreender a minha ausência durante o período letivo e que muito contribuíram com conselhos e palavras de motivação.

Aos meus colegas de trabalho, pelo estímulo e apoio durante o desenvolvimento deste projeto.

À Christiane Peixoto, pelo carinho e dedicação durante o desenvolvimento deste trabalho.

E por fim, agradeço aos amigos que adquiri através desta jornada acadêmica: Alessandra Corrêa, Felipe Moreno e Regina Lorenzi que, sem dúvida, foram imprescindíveis para chegada ao final deste percurso.

"Há momentos difíceis na vida. Grandes ou pequenas, as dificuldades podem ser decisivas. Somente a firme determinação de enfrentar as adversidades leva o indivíduo a vencê-las verdadeiramente. Nessas horas cruciais, jamais hesite o mínimo."

Dr. Daisaku Ikeda

RESUMO

Atualmente, muitos estudos demonstram que os projetos não obtêm sucesso na grande maioria das vezes, seja porque os gastos superam o orçamento, o tempo supera o cronograma, as funcionalidades não resolvem os problemas do cliente e a baixa qualidade dos *softwares* não contribui ou até mesmo todos estes fatores estão presentes no mesmo projeto. Esta monografia propõe a adoção do conjunto formado pelas práticas e valores de três metodologias: o PMBOK, o Extreme Programming (XP) e o SCRUM para que, juntas, possam fornecer um mecanismo e estrutura eficazes para garantir e elevar as chances de sucesso em diversos projetos de *software*. Para validação dessa monografia, conduziu-se uma revisão da literatura e um estudo de caso é apresentado, no qual é aplicada a metodologia criada a partir das modificações realizadas no PMBOK, no Extreme Programming (XP) e no SCRUM. Essa metodologia foi utilizada durante o período de três anos em projetos comerciais, com resultados positivos.

Palavras-Chave: Metodologia. Processo. Gestão. Qualidade.

ABSTRACT

Currently, many studies show that the projects in most of the cases are not successful, either because the costs exceed the budget, the time exceeds the schedule, the features do not solve customer problems and the low quality of *software* does not contribute or even even all these factors can be present in the same project. This study proposes the adoption of a combination of the practices and three methodologies values: PMBOK, Extreme Programming (XP) and SCRUM, these methodologies together can provide an effective mechanism and structure to ensure and increase the chances of success in various *software* projects. It was conducted a literature review for this study validation and a case study is presented, which is applied the methodology created from the changes made to the PMBOK, the Extreme Programming (XP) and SCRUM. This methodology was used during the period of three years in commercial projects, with positive results.

Keywords: Methodology. Process. Management. Quality.

LISTA DE GRÁFICOS

Figura 1	– Fases do desenvolvimento da nova metodologia.	50
Gráfico 1	– Demissionários.	70
Gráfico 2	– Gráfico do Desenvolvimento dos projetos antes, durante e após a implantação da nova metodologia.	71
Gráfico 3	– Não conformidade por ano.	73

LISTA DE QUADROS

Quadro 1	– RSI – Relatório de Solicitação Inicial.	51
Quadro 2	– AVT – Análise de Viabilidade Técnica.	52
Quadro 3	– Planejamento do Projeto.	53
Quadro 4	– Desenvolvimento do Projeto.	55
Quadro 5	– Teste e Qualidade do Produto.	56
Quadro 6	– Homologação do Projeto.	58
Quadro 7	– Conclusão do Projeto.	59
Quadro 8	– Gerenciamento de atividades.	60

LISTA DE ABREVIATURAS E SIGLAS

ALM	<i>Application Lifecycle Management</i>
AVT	Análise de Viabilidade Técnica
GMUD	Gerência de Mudanças
PMBOK	Project Management Body of Knowledge
PKI	<i>Public key infrastructure</i>
PMI	<i>Project Management Institute</i>
QA	<i>Quality Assurance</i>
RSI	Relatório de Solicitação Inicial
RUP	<i>IBM Rational Unified Process</i>
SCRUM	<i>Framework</i>
XP	Extreme Programming

SUMÁRIO

1. INTRODUÇÃO	14
1.1 OBJETIVO	16
1.2 ESCOPO	16
1.3 ESTRUTURAS DOS CAPÍTULOS.....	17
2. METODOLOGIAS	18
2.1 PROJECT MANAGEMENT BODY OF KNOWLEDGE (PMBOK).....	20
2.2 SCRUM	26
2.2.1 BACKLOG	27
2.2.2 SPRINT	27
2.2.3 DAILY SCRUM.....	27
2.2.4 BACKLOG DO SPRINT.....	29
2.2.5 SCRUMMASTER	29
2.2.6 PRODUCT OWNER.....	30
2.3 EXTREME PROGRAMMING (XP).....	31
2.3.1 VALORES DO EXTREME PROGRAMMING (XP)	33
2.3.2 PRINCÍPIOS DO EXTREME PROGRAMMING (XP)	35
2.3.3 PRÁTICAS DO EXTREME PROGRAMMING (XP)	37
2.3.4 CICLO DE VIDA DE UM PROJETO EXTREME PROGRAMMING	43
2.3.5 PAPÉIS DENTRO DO PROJETO USANDO EXTREME PROGRAMMING (XP)	45
3. ESTUDO DE CASO	48
3.1. INTRODUÇÃO	48
3.2. METODOLOGIA UTILIZADA	48
3.2.1. DESENVOLVIMENTO DE PROJETO	50
3.2.2. TAMANHO DOS PROJETOS.....	50
3.3. IMPLEMENTAÇÃO DA METODOLOGIA.....	64
3.4. IDENTIFICAÇÃO DOS PROBLEMAS.....	65
3.5. CONSIDERAÇÕES FINAIS	67
4. MÉTRICAS.....	68
4.1. PESSOAS	68
4.2. PROJETOS.....	70

5. QUALIDADE	72
6. CONCLUSÃO	73
7. REFERÊNCIAS BIBLIOGRÁFICAS	75

1. INTRODUÇÃO

Realizar uma boa gestão de processos no ramo empresarial é uma tarefa árdua. Durante muito tempo, buscou-se um método adequado e corretivo para auxiliar na eficácia e efetividade da gestão de processos.

Ocorre que, muitas vezes, a empresa tenta se adaptar aos termos da metodologia escolhida. Contudo, o procedimento correto é justamente o inverso, no qual a metodologia é que deve ser adaptada à cultura e aos costumes da corporação, para que, assim, possa alcançar o resultado buscado pela companhia.

Quando esse procedimento não ocorre de maneira correta – no qual a metodologia se molda à empresa – os objetivos almejados não são atingidos, o que, consequentemente, resulta em prejuízos em grande escala.

Com a ausência de uma boa gestão de processos, moldada aos costumes da empresa, é possível identificar que os riscos apresentados são, na maioria das vezes, decorrentes de gastos superiores ao orçamento e de prazos que ultrapassam o cronograma traçado para a finalidade em questão. As funcionalidades, na maioria das vezes, não atingem a problemática apontada pelo cliente, juntamente com a falta de qualidade dos *softwares* apresentados, que não agradam o público alvo.

O conjunto negativo dos fatores acima ilustrados comprometem e denigrem a imagem da organização.

Atualmente, existem diversas metodologias no mercado que possuem, como escopo auxiliar, a gestão de processos em uma empresa. Contudo, o presente trabalho irá adentrar, ainda que superficialmente, nas metodologias SCRUM, Extreme Programming (XP) e PMBOK, pois estas servirão como base para o estudo de caso ora apresentado.

Este estudo de caso será realizado através da empresa Certified Inovation S.A – nome fantasia – que, em busca do aperfeiçoamento e melhoramento de resultados voltados à área de gestão de processos, realizou uma pesquisa dentro da própria corporação, durante três anos, para que, juntamente com as metodologias SCRUM, Extreme Programming (XP) e PMBOK, fosse possível adaptar o mecanismo dessas metodologias aos procedimentos da empresa, visando alcançar um melhor aproveitamento e adaptação dos seus colaboradores.

Assim, compõe o desafio deste estudo, inicialmente, introduzir extrinsecamente as metodologias SCRUM, Extreme Programming (XP) e PMBOK, seguidos de um breve conceito e aplicabilidade destas, para que, posteriormente, seja devidamente comprovado, com base no estudo de caso, que embora estas práticas sejam úteis isoladamente, quando são devidamente adaptadas a um caso concreto, podem potencializar a sua eficácia.

Por fim, com este trabalho é possível observar que há a viabilidade na criação de uma própria metodologia a partir de outras já existentes no mercado, uma vez que esta seja compatível com a cultura da organização.

1.1 OBJETIVO

O presente trabalho possui como objetivo central apresentar um estudo de caso que demonstre a funcionalidade e viabilidade de se adaptar uma ou várias metodologias visando auxiliar o gerenciamento de processos de uma empresa.

Sendo assim, após um estudo aprofundado no interior de uma grande corporação multinacional, foram analisados os costumes e a cultura da empresa, bem como a identificação dos problemas existentes – passíveis de aprimoramento –, para que fosse definido o processo adequado que mais atingisse os objetivos da referida corporação.

Após esta minuciosa identificação da necessidade corporativa e da definição de qual o melhor procedimento a ser seguido, foram implementadas as metodologias dentro da empresa, seguidas de adequações e adaptações realizadas de acordo com as problemáticas apresentadas.

Posteriormente a este processo de aplicação e adaptação das metodologias, é incontestável que houve inúmeros avanços e aperfeiçoamentos realizados em lapso temporal de curto prazo, os quais serão metricamente demonstrados no capítulo pertinente, deixando certo que é possível aprimorar e potencializar a gestão de processos baseada em metodologias já existentes, adaptáveis a um caso concreto.

1.2 ESCOPO

Busca-se, com a presente monografia, demonstrar a eficácia das metodologias aplicadas à uma determinada empresa, as quais foram adaptadas e moldadas de acordo com as necessidades e costumes da companhia.

Para tanto, foi utilizado como base acadêmica, uma renomada empresa multinacional conceituada por sua atuação no ramo de certificação digital, que para o presente estudo, apresentará o nome fantasia de "Certified Inovation S.A.".

A "Certified Inovation S.A.", foi criada em meados de 2011 e possui, como principal atividade a criação de *softwares* ocupando, atualmente, a terceira colocação no ramo em que atua, em decorrência do excelente trabalho realizado, envolvendo a gestão de processos.

Contudo, antes de ser uma das empresas mais respeitadas nesta área, a "Certified Inovation S.A." enfrentou uma crise na execução dos projetos, que a levou a

reestruturação dos processos e procedimentos internos para poder alcançar sua ascensão no mercado.

O método utilizado, que resultou em significativa lucratividade da empresa, conseqüentemente, a diminuição de gastos, deu-se através da implantação de metodologias que auxiliaram o gerenciamento do negócio.

Essas metodologias são consideradas como importantes ferramentas que sistematizam os processos internos de uma organização. E, após a implantação destas metodologias – de acordo com a cultura da empresa –, foi de extrema relevância a realização do devido treinamento e conscientização dos colaboradores acerca dos novos procedimentos seguidos, e principalmente, para àqueles voltados às áreas de planejamento e execução dos projetos implantados.

Por fim, esta monografia irá ilustrar, com fatos e dados reais, que o caminho para um bom desempenho de uma empresa ocorre através da estruturação dos processos internos, de acordo com a sua própria cultura, inicialmente, compreendendo a real situação da empresa, para que, posteriormente, haja o devido planejamento e intervenções de melhoria contínua.

1.3 ESTRUTURAS DOS CAPÍTULOS

Este trabalho está dividido em cinco capítulos.

Em um primeiro momento, é realizada uma breve síntese deste trabalho, apresentando as problemáticas identificadas na área de *software*, especificamente, pela ausência de um processo definido e suas desvantagens.

No segundo capítulo, serão introduzidas, superficialmente, as metodologias disponíveis no mercado, destacando aquelas utilizadas no estudo de caso, dando ênfase às suas particularidades.

No capítulo seguinte, é apresentado o estudo de caso propriamente dito, juntamente com os percalços enfrentados pela empresa em questão, antes de utilizarem uma metodologia moldável às suas culturas e costumes.

Já no quarto capítulo, são evidenciadas as métricas que foram obtidas antes, durante e após a implantação da metodologia na companhia em questão. Para tanto, com o intuito de facilitar a compreensão das estatísticas apresentadas para o caso em conteúdo, haverá a seguinte subdivisão: pessoas, projetos e qualidade.

Por fim, o último capítulo será destinado à conclusão do presente trabalho, com um panorama geral do estudo de caso, demonstrando a eficácia das práticas adotadas e da funcionalidade e viabilidade das metodologias PMBOK, Scrum e Extreme Programming (XP), juntamente com as melhorias e aprimoramentos que foram proporcionados à empresa em apreço.

2. METODOLOGIAS

Conceitualmente e simplificadamente, a palavra metodologia pode ser definida como um ramo da lógica que se ocupa dos métodos das diferentes ciências. Pode-se dizer que este conceito é bem mais amplo do que um simples conjunto de métodos utilizados para alcançar um determinado objetivo, mas que também abrange os próprios fundamentos, pressupostos, regras e diligências necessárias para se atingir determinado fim.

As metodologias as quais serão estudadas abaixo, referem-se a procedimentos voltados para a gestão e planejamento de projetos corporativos. Segundo Martins (2007, p. 2), a metodologia descreve as atividades envolvidas na definição, construção e implementação de um sistema; a metodologia é um *framework*.

Já Nogueira (2004), defende que com a implementação de uma metodologia, a empresa produtora de *software* terá mais controle e gerenciamento através de uma metodologia focada aos processos fundamentais de desenvolvimento de *software*. Exemplos: RUP, XP e ALM.

Martins (2007, p.2), na mesma obra supracitada, defende que há a existência de micro e macro componentes: Os macros componentes definem o fluxo geral e a sequência de trabalho, enquanto que os microcomponentes incluem as regras gerais de projetos técnicos (*design*), padrões (*patterns*) e boas práticas.

Para que as metodologias possam ser implementadas em uma empresa, é necessário que seja realizado um projeto, com todas as problemáticas que deverão ser superadas, determinação de prazo para o término de cada etapa, previsão orçamentária, execução de processos e procedimentos que deverão ser seguidos, entre outros quesitos. Ou seja, deve haver uma prévia estruturação e planejamento interno para que o projeto possa ser concluído atendendo, com excelência, o objetivo principal.

Segundo o PMBOK (PMI, 2014) "projeto é um esforço temporário empreendido para criar um produto, serviço ou resultado único".

Genericamente, "projeto" significa "empreendimento", e como tal é um trabalho que visa a criação de um produto ou a execução de um serviço específico, temporário, não repetitivo e que envolve certo grau de incerteza na realização (MARTINS, 2010). Para entender de gestão de projetos, em primeiro lugar é preciso saber reconhecer o que é um projeto. Trata-se de um empreendimento com objetivo bem definido, que consome recursos e opera sob pressão de prazos, custos e qualidade (KERZNER, 2007, p.15).

As principais características dos projetos são a temporariedade, a individualidade do produto ou serviço a ser desenvolvida pelo projeto, a complexidade e a incerteza (VARGAS, 2007, p. 6).

A área de Gerenciamento de Projetos está ganhando grande enfoque nas áreas corporativas, pela possibilidade de se ter um ramo especializado para gerenciar os mais diversos projetos que são implantados em uma empresa.

Como se verá a seguir, no estudo de caso, antes de ser implantado uma ou diversas metodologias em uma corporação é necessário que haja a elaboração de um projeto, para que, posteriormente, a metodologia seja implantada e adequada às necessidades da corporação.

Apenas por meio do monitoramento, controle e aperfeiçoamento de processos da organização é que esta poderá aperfeiçoar seu desempenho ao longo do tempo. Desta forma, sistemas de informação adequadamente construídos e gerenciados conduzem as organizações a se tornarem cada vez mais eficazes e eficientes no cumprimento de sua missão (FERNANDES; RALHA, 2005).

Dentre as diversas metodologias disponíveis no mercado, será dado destaque apenas para aquelas utilizadas neste estudo de caso – apresentado no Capítulo 3, deste trabalho. Sendo assim, abordaremos, no próximo item as três estruturas que foram implementadas no estudo de caso: SCRUM, Extreme Programming (XP) e PMBOK, que serão devidamente embasadas a um conceito doutrinário, seguidas com os meios e formas de aplicabilidade e implementação.

2.1 PROJECT MANAGEMENT BODY OF KNOWLEDGE (PMBOK)

O PMBOK (Project Management Body of Knowledge) é uma das metodologias mais conhecidas no mundo na área de gerenciamento de projetos. O PMBOK é a base de conhecimento do PMI (*Project Management Institute*), sendo seguida mundialmente como um padrão de metodologia, devido à sua divulgação e penetração em praticamente todo o mundo. Pode-se afirmar que o PMBOK é um conjunto de processos e práticas na área de gerenciamento de projetos.

Segundo o PMBOK:

A aceitação do gerenciamento de projetos como uma profissão indica que a aplicação do conhecimento, processos, habilidade, ferramentas e técnicas pode ter um impacto significativo no sucesso do projeto. O Guia PMBOK indica esse subconjunto do conhecimento em gerenciamento de projeto que é amplamente reconhecido como boa prática. "Amplamente reconhecido" significa que o conhecimento e as práticas descritas são aplicáveis à maioria dos projetos na maior parte das vezes e que um consenso em relação ao seu valor de utilidade. "Boa Prática" significa que existe um consenso geral de que a aplicação do conhecimento, habilidades, ferramentas e técnicas pode aumentar as chances de sucesso de muitos projetos. "Boa Prática" não significa que o conhecimento descrito deva ser sempre aplicado uniformemente a todos os projetos; organização e/ou a equipe de gerenciamento do projeto é responsável por determinar o que é apropriado para um projeto específico (PMI, 2014).

Ao aplicar os conceitos de controle de processos de construção civil à engenharia de *software*, os métodos podem ser classificados em duas modalidades: metodologia com processos definidos (ou prescritivos) e a metodologia empírica.

A primeira define o contexto do projeto e o escopo das entregas logo no início do projeto, enquanto que a outra define as entregas de forma abrangente e superficial, começando por um contexto inicial, que depois evolui (MARTINS, 2007, p.2).

As organizações sempre querem atingir ou até mesmo exceder as expectativas de seus clientes, isso envolve, sem dúvida, um equilíbrio entre as demandas que são concorrentes:

- Escopo, prazo, custo e qualidade.

Para que haja esse equilíbrio é necessário que esse processo seja devidamente seguido. Assim, é possível saber o direcionamento do projeto.

O gerenciamento de projeto é aplicável a diversas áreas, conforme dispõe Vargas (2007, p. 5). Atualmente, é possível destacar as seguintes áreas de aplicabilidade como os principais utilizadores da técnica de gerenciamento de projetos:

- Administração de empresas;
- Desenvolvimento de programas de computadores;
- Engenharia e construção civil;
- Estratégia militar;
- Manutenção de plantas e equipamentos;
- *Marketing* e publicidade e,
- Pesquisa e desenvolvimento.

O gerenciamento de projetos (PMI, 2014) é a aplicação do conhecimento, juntamente com as ferramentas e técnicas, são as atividades do projeto para atender aos seus requisitos.

Segundo PMI (2014), os processos de gerência de projetos podem ser organizados em:

- **Grupo de processos de iniciação** – iniciar um projeto ou fase obtendo as autorizações necessárias e se comprometer a executá-la;
- **Grupo de processos de planejamento** – planejar e refinar os objetivos, seguindo um esquema de trabalho para conseguir obter êxito nas metas para os quais resultaram na criação do projeto;
- **Grupo de processos de execução** – processo para execução do projeto, coordenar pessoas e outros recursos para obter êxito do planejamento;
- **Grupo de processos de monitoramento** – realizar o acompanhamento através do monitoramento dos objetivos do processo, avaliar o progresso e administrar ações para alterar quando for necessário;
- **Grupo de processos de encerramento** – formalização do fim do projeto ou fase, de forma estruturada e organizada.

Verzuh (1999, p. 25), propõe uma estruturação dos projetos por meio dos processos "definição, planejamento, execução e encerramento" e menciona que estes se repetem ao longo dos vários estágios do ciclo de vida do projeto e do produto. Maximiano (2002, p. 49), compactua com a mesma ideologia ao defender que os processos da administração de projetos – planejamento, organização, execução e controle – são necessários para o projeto como um todo e para cada fase de seu ciclo de vida.

Segundo o PMBOK, (PMI, 2014), para se obter sucesso, um projeto deve seguir dez áreas de gerenciamento de projetos, são elas:

- Gerenciamento da Integração;
- Gerenciamento de Escopo;
- Gerenciamento de Custos;
- Gerenciamento de Qualidade;
- Gerenciamento das Aquisições;
- Gerenciamento de Recursos Humanos;
- Gerenciamento das Comunicações;
- Gerenciamento de Risco;
- Gerenciamento de Tempo, e
- Gerenciamento das Partes Interessadas.

As áreas acima elencadas são consideradas essenciais dentro do guia PMBOK e também pelo gerenciamento de projetos, pois possuem o intuito de focar e delinear os objetivos e recursos, bem como a programação de cada projeto.

Esses conhecimentos foram unificados e moldados para obtenção do sucesso da pesquisa do estudo de caso, no qual a empresa Certified Innovation S.A. obteve êxito com a utilização da nova metodologia.

2.1.1 GERENCIAMENTO DA INTEGRAÇÃO DO PROJETO

O gerenciamento da integração do projeto inclui os processos e as atividades para identificar, definir, combinar, unificar, e coordenar os vários processos e atividades dentro do grupo de processos de gerenciamento do projeto (PMI, 2014).

Processo é um conjunto de atividade, ações e tarefas realizadas na criação de algum produto de trabalho (PRESSMAN, 2009. p.40).

2.1.2 GERENCIAMENTO DO ESCOPO DO PROJETO

O gerenciamento do escopo do projeto inclui os processos necessários para assegurar que o projeto inclui todo o trabalho necessário, e apenas o necessário, para terminar o projeto com sucesso (PMI, 2014).

Segundo o KERZNER (2006, p. 406), O escopo é a soma de todas as entregas necessárias como parte do projeto. Isso inclui todos os produtos, serviços e resultados.

2.1.3 GERENCIAMENTO DO TEMPO DO PROJETO

O gerenciamento do tempo inclui os processos necessários para gerenciar o término pontual do projeto (PMI, 2014).

O gerenciamento do tempo é uma das áreas mais visíveis do gerenciamento do projeto, isso porque um dos principais focos daqueles que se envolvem com projetos é o controle de prazos e montagem de cronograma. O principal objetivo desta área é o controle para que não haja atraso e o projeto seja finalizado pontualmente (Vargas, 2009).

2.1.4 GERENCIAMENTO DOS CUSTOS DO PROJETO

Segundo BROD, o gerenciamento dos custos do projeto inclui os processos envolvidos em planejamento, estimativa, orçamento, financiamento, gerenciamento e controle de custos, de modo que o projeto possa ser terminado dentro do orçamento aprovado.

Kent pôde observar que o custo de desenvolvimento de um sistema estava muito mais ligado ao custo das pessoas do que ao hardware ou às licenças dos *softwares* empregados, e que, para a boa continuidade e o crescimento de um projeto, era necessário envolver o usuário (Cliente) (BROD, 2013).

O principal objetivo do gerenciamento dos custos é planejar e controlar a parte financeira do projeto e assegurar que o orçamento do projeto seja o suficiente para obter todos os recursos necessários para o projeto do mesmo (Vargas, 2009).

2.1.5 GERENCIAMENTO DA QUALIDADE DO PROJETO

O gerenciamento da qualidade do projeto inclui os processos e as atividades da organização executora que determina as políticas de qualidade, os objetivos e as responsabilidades, de modo que o projeto satisfaça às necessidades para as quais foi empreendido (PMI, 2014).

O objetivo dessa área de conhecimento é garantir que o projeto terá a qualidade desejada, assegurando a satisfação das partes interessadas (Vargas, 2009).

2.1.6 GERENCIAMENTO DOS RECURSOS HUMANOS DO PROJETO

O gerenciamento dos recursos humanos do projeto inclui os processos que organizam, gerenciam e guiam a equipe do projeto. A equipe do projeto consiste das pessoas com papéis e responsabilidades designadas para completar o projeto (PMI, 2014).

De acordo com Vargas (2009), o gerenciamento de recursos humanos tem como principal objetivo otimizar a utilização dos indivíduos envolvidos no projeto. As pessoas são os recursos mais importantes de um projeto, pois definem as metas, os planos, produzem resultados e controlam as atividades. Os resultados do projeto são fruto das relações humanas e habilidades interpessoais dos envolvidos, já que a motivação dos profissionais está cada vez mais ligada com a satisfação pessoa e qualidade de vida.

2.1.7 GERENCIAMENTO DAS COMUNICAÇÕES DO PROJETO

O gerenciamento das comunicações do projeto inclui os processos necessários para assegurar que as informações sejam planejadas, coletadas, criadas, distribuídas, armazenadas, recuperadas, gerenciadas, controladas, monitoradas e finalmente dispostas de maneira oportuna e apropriada (PMI, 2014).

É importante que haja um processo de comunicação efetiva para garantir que as informações necessárias cheguem às pessoas corretas no tempo certo. Este processo é utilizado para que a equipe trabalhe de maneira integrada, auxiliando na solução de problemas e identificação de oportunidades (VARGAS, 2009).

2.1.8 GERENCIAMENTO DOS RISCOS DO PROJETO

O gerenciamento dos riscos do projeto inclui os processos de planejamento, identificação, análise, planejamento de respostas e controle de riscos. Os objetivos do gerenciamento dos riscos do projeto são aumentar a probabilidade e o impacto dos eventos positivos e reduzir a probabilidade e o impacto dos eventos negativos no projeto (PMI, 2014).

2.1.9 GERENCIAMENTO DAS AQUISIÇÕES DO PROJETO

O gerenciamento das aquisições do projeto inclui os processos necessários para comprar ou adquirir produtos, serviços ou resultados externos à equipe do projeto. A organização pode ser tanto o comprador quando o vendedor dos produtos, serviços ou resultados de um projeto (PMI, 2014).

O gerenciamento de aquisições tem como meta garantir que todo o elemento externo ao projeto forneça seu produto ou serviço. A relação entre o fornecedor e o projeto é geralmente determinada pelo risco incorrido entre eles e, por causa deste risco, é necessário avaliar não só o custo, mas também o risco de estabelecimento de um contrato entre as partes (VARGAS, 2009).

2.1.10 GERENCIAMENTO DAS PARTES INTERESSADAS DO PROJETO

O gerenciamento das partes interessadas do projeto inclui os processos exigidos para identificar todas as pessoas, grupos ou organização que podem impactar ou serem impactados pelo projeto, analisar as expectativas das partes interessadas e seu impacto no projeto, e desenvolver estratégias de gerenciamento apropriados para o engajamento eficaz das partes interessadas nas decisões e execução do projeto (CANEDO, 2007).

O gerenciamento de projetos bem-sucedido inclui o gerenciamento ativo dessas interações para atender satisfatoriamente às necessidades do patrocinador, do cliente e de outras partes interessadas. Esta norma descreve a natureza dos processos de gerenciamento de projetos em termos da integração entre os processos, das interações dentro deles e dos objetivos a que atendem (CANEDO, 2007).

Neste capítulo foi possível demonstrar as diversas áreas da gerência de projetos seguindo os parâmetros apontados não só pelo guia PMBOK, mas também, de acordo com as considerações de outros autores sobre o tema.

Sendo assim, ainda neste item, foi abordado de forma clara e didática, os aspectos imprescindíveis e com relevância para que ocorra a implantação dos processos dentro da organização de forma segura e eficaz.

2.2 SCRUM

O SCRUM foi concebido para o gerenciamento de projetos em uma fábrica de automóveis e produtos de consumo.

A primeira referência que se teve notícia deu-se através de artigo apresentado na Universidade *Harvard* em 1986, por Ikujiro Nonaka e Hirotaka Takeuchi.

No artigo, os autores destacavam que em projetos com equipes menores, interações, valores e organização havia melhores resultados.

Em 1995, o SCRUM teve sua definição exposta por Jeff Sutherland e Ken Schwaber, que trabalharam juntos para consolidá-lo como método de desenvolvimento de *software*, ficando conhecido por todo o mundo sob a denominação SCRUM. Juntos, Sutherland e Schwaber publicaram um artigo em que compartilharam algumas de suas experiências com a implementação do SCRUM nas corporações, bem como os resultados obtidos.

A metodologia SCRUM segue os princípios do Manifesto Ágil (MANIFESTO ÁGIL, 2001) e possui como representantes do método: Mike Beedle, Ken Schwaber e Jeff Sutherland.

Apenas para título de conhecimento, o nome SCRUM surgiu da comparação entre desenvolvedores e jogadores de *Rugby*. SCRUM é a denominação da rápida reunião que ocorre quando os jogadores de *Rugby* vão iniciar um lance.

2.2.1 BACKLOG

O ponto inicial do SCRUM é o *Backlog* do Produto, uma importante etapa considerada a prática responsável pelo armazenamento e gerenciamento dos requisitos coletados do cliente. É uma relação (lista) que deverá conter todas as funcionalidades desejadas para um produto.

O *Backlog* não precisa, necessariamente, estar completo no início do projeto.

Em outras palavras, é uma lista contendo todas as características, funções, tecnologias, melhorias e correções que constituem a versão futura do produto a ser entregue.

Conforme Marçal (2007), o objetivo é organizar as atividades dos itens do *Sprint Backlog* em quatro estados: para fazer, em andamento, para verificar e concluído.

Assim, o *Backlog* do Produto é uma relação de funcionalidades, ordenadas por prioridade, que provavelmente serão desenvolvidas durante o projeto.

2.2.2 SPRINT

O *Sprint* é o período de tempo no qual são executados os itens de trabalho definidos no *Backlog* do Produto pela equipe, em outras palavras, são os requisitos já definidos.

O *Sprint* normalmente possui a duração de quatro semanas – não é uma regra, podendo ser alterado conforme as necessidades específicas do caso. Conforme Kniberg (2007, p. 25): O propósito da reunião de planejamento do *Sprint* é dar à equipe informação suficiente para trabalharem em paz por algumas semanas, e para dar ao *product owner* confiança suficiente para deixá-los fazerem isto.

No *Sprint*, as equipes que decidem a duração a ser adotada para a execução do projeto. No caso do desenvolvimento de *software*, o *Sprint* inclui as fases tradicionais do desenvolvimento de *software*: requisitos, análise, projeto e entrega.

2.2.3 DAILY SCRUM

O *Daily SCRUM* é a reunião diária que deve ser realizada para ajustar os procedimentos pertinentes ao projeto.

É o momento em que equipe deve se juntar, diariamente, para uma reunião de 15 minutos no máximo – sem exceder este tempo –, com todos os membros em pé, de forma a conseguir maior agilidade.

Este encontro diário ocorre entre os membros da equipe para definir as tarefas que serão executadas no decorrer do dia, e também para se tomar nota dos resultados das atribuições, realizadas ou não, do dia anterior.

A ideia central desta reunião é que ocorra sempre no mesmo horário e local, durante o *Sprint* e que tenha como objetivo a respostas de três perguntas de cada membro envolvido no projeto, a explicação deve ser breve e objetiva, são elas:

- O que foi feito ontem?
- O que será feito hoje?
- Há algum obstáculo para a execução das atividades?

De acordo com Schwaber e Sutherland existem três pontos para inspeção e adaptação em SCRUM:

A *Daily Scrum* é uma reunião utilizada para inspecionar o progresso em direção à Meta da *Sprint* e para utilizar adaptações que otimizem o valor do próximo dia de trabalho.

Além disso, as reuniões de Revisão da *Sprint* e de Planejamento da *Sprint* são utilizadas para inspecionar o progresso em direção à Meta da *Release* e para fazer as adaptações que otimizem o valor da próxima *Sprint*.

Finalmente a Retrospectiva da *Sprint* é utilizada para revisar a *Sprint* passada e definir que adaptações tornaram a próxima *Sprint* mais produtiva, recompensadora e gratificante (SCHWABER E SUTHERLAND, 2010, p. 4).

Na reunião diária de Scrum, os membros da equipe não respondem os questionamentos como forma de cobrança, mas sim como forma de alinhar a execução das atividades como um todo, gerando ainda mais comprometimento com o projeto. Desta forma, todos os membros ficam conscientizados sobre o *status* de cada tarefa e podem, inclusive, auxiliar em conjunto caso exista algum empecilho no decorrer do *Sprint*.

2.2.4 BACKLOG DO SPRINT

O *Backlog* do *Sprint* é um subconjunto do *Backlog* do Produto.

Pode-se dizer que é uma lista de atividades a serem desenvolvidas durante o *Sprint* e sua definição acontece durante a reunião de Planejamento do *Sprint*.

De acordo com Kniberg, sobre os *Sprints*:

Bem, *sprints* curtos são bons. Eles permitem que a equipe seja “ágil”, isto é, mude de direção frequentemente. *Sprints* curtos = ciclo curto de *feedback* = entregas mais frequentes = *feedback* mais frequente do cliente = menos tempo perdido, indo na direção errada = aprender e melhorar rápido, etc. Entretanto, *sprints* longos são bons também. A equipe tem mais tempo para ganhar ritmo, ela tem mais espaço para se recuperar dos problemas, e conseguir atingir o objetivo do *Sprint*, você tem menos overhead em termos de reunião de planejamento, apresentações, etc. (KNIBERG 2007, p. 25)

No início do projeto, cliente, desenvolvedor, analista de sistema e de qualidade, e arquiteto de *softwares* definem o *Backlog* do Produto (lista de requisitos). Nesse momento, serão estimados os custos do projeto e definidas as datas de entrega a partir da priorização realizada nesta reunião. Também é feita uma análise inicial de riscos do projeto. E, por fim, as ferramentas de trabalho que serão necessárias para execução do projeto, bem como, os integrantes das equipes que são selecionados de acordo com a experiência, expertise e criticidade do projeto.

Um dos desenvolvedores é eleito como *Scrummaster*, cujo papel se assemelha a um gerente de projetos, vale lembrar que o *Scrummaster* pode ser eleito a cada projeto ou ser um cargo fixo, depende da forma que cada empresa trabalha. Um ponto importante a ser ressaltado é que existem diferenças cruciais entre um “Gerente de Projetos” – como já discutiremos em momentos anteriores e um *Scrummaster* – definição que será especificada a seguir.

2.2.5 SCRUMMASTER

O *Scrummaster* trabalha gerenciando os processos da metodologia *Scrum*.

Com essa gestão, este responsável observa o andamento do projeto e evita que existam impedimentos para que os membros da equipe realizem seu trabalho. Segundo os autores Schwaber e Sutherland, o *Scrummaster*:

O *Scrummaster* pode ser um membro do time; por exemplo, um desenvolvedor realizando tarefas da *Sprint*. No entanto, isso frequentemente leva a conflitos quando o *Scrummaster* deve escolher entre remover impedimentos e realizar tarefas. O *Scrummaster* nunca deve ser o *product owner* (SCHWABER e SUTHERLAND, 2010, P. 6).

Remover os obstáculos apontados na reunião de *Scrum* diária é seu dever, de modo que a equipe de desenvolvimento se concentre apenas em questões técnicas do projeto. Obstáculos são colocados em uma lista chamada de *Backlog* de Impedimentos, que deve ficar em local visível por todos os membros do projeto.

2.2.6 PRODUCT OWNER

É o do Dono do Produto (*Product Owner*). Este membro da equipe geralmente representa o cliente (interno ou externo).

O *Product Owner* define quais são os requisitos e qual é o grau de importância e prioridade de cada um deles.

Schwaber e Sutherland definem este profissional:

O *product owner* pode ser um membro do time, trabalhando também em desenvolvimento. Mas essa responsabilidade adicional pode reduzir a sua habilidade de lidar com as partes interessadas. No entanto, o *product owner* nunca pode ser o *Scrummaster* (SCHWABER e SUTHERLAND, 2010, p. 6).

Como cliente, o *product owner*, deve analisar quais funcionalidades foram entregues em cada *Sprint*, além de conhecer muito bem as regras que regem àquele negócio, para que possa sanar eventuais questionamentos e dúvidas futuras que possam surgir em relação as funcionalidades do produto.

Ao iniciar cada *Sprint*, quando a equipe faz a lista das atividades que precisam ser realizadas no primeiro *Sprint* (*Backlog* do *Sprint*), as responsabilidades e atribuições serão divididas entre os membros da equipe. Os desenvolvedores, analistas,

arquitetos e equipe de qualidade irão discutir os padrões que serão adotados, as fases de análise, a construção do código e os testes se iniciam.

Ao final de cada *Sprint*, uma versão do produto é apresentada ao cliente para obter a aprovação. Nesse caso, o cliente realiza uma homologação do sistema e identifica se os requisitos listados estão em conformidade com o acordado na reunião.

Teles (2014, p. 64), aduz a importância da participação ativa do cliente no desenvolvimento do projeto: A participação do cliente viabiliza a simplicidade do processo. Além disso, ela permite que o projeto seja conduzido por meio de uma série de pequenos ajustes e não por meio de mudanças bruscas ao longo do caminho.

A quantidade de funcionalidades não entregues, a necessidade de alterações para corrigir as “não-conformidades” encontradas ou para atualização tecnológica, os problemas técnicos encontrados, os riscos e as estratégias para evitá-los são exemplos de controles observados durante o desenvolvimento.

2.3 EXTREME PROGRAMMING (XP)

A metodologia Extreme Programming (XP) é resultado da experiência de um projeto realizado no interior de empresa multinacional fabricante de automóveis: a *Chrysler Corporation*.

Este projeto possuía o objetivo de construir um sistema voltado para a área financeira, que auxiliasse a elaborar a folha de pagamento. A empresa já tinha utilizado outras metodologias em situações anteriores e não tinha logrado êxito.

Após a implementação da metodologia Extreme Programming (XP) a empresa conseguiu suprir suas necessidades. Com o sucesso do projeto, a metodologia começou a ter destaque no âmbito acadêmico, empresarial e, também, nos setores de tecnologia de *software*.

Segundo Beck (2004) as práticas de XP, em sua essência, vêm sendo utilizadas muito antes de sua criação. Porém, muitas foram reunidas e substituídas por práticas mais complexas ao longo do tempo. De acordo com o entendimento do autor:

Em 1990, Kent Beck começou a estudar, em bases práticas, o que facilitava e o que dificultava a produção de *software*, mas sem olhar os programas

que eram descritos, e sim mantendo o foco na satisfação do cliente e na forma como as equipes de desenvolvedores trabalhavam. Kent pôde observar que o custo de desenvolvimento de um sistema estava muito mais ligado ao custo das pessoas do que do *hardware* ou às licenças dos *softwares* empregados, e que, para a boa continuidade e o crescimento de um projeto, era necessário envolver o usuário (cliente) no processo de produção e tornar os programas tão claros quanto o possível para a leitura por “humanos”. Por isso Kent desenvolveu a metodologia XP com base em quatro princípios básicos: comunicação, simplicidade, realimentação (*feedback*) e coragem. A partir de 1996, Kent Beck passou a aplicar seus conceitos na *Daimler-Chrysler* (BROD, 2013, P. 36).

Esta metodologia serve como um importante instrumento para inúmeras pesquisas e discussões, possuindo como grandes defensores e idealizadores os nomes mais renomados da área de engenharia e programação de *software*: Kent Beck, Ron Jeffries, Martin Fowler e Grady Booch.

O grande avanço na popularidade do Extreme Programming (XP) deu-se em decorrência do histórico positivo de projetos e relatos de empresas que utilizaram esta metodologia e conseguiram atingir suas metas.

Consequentemente, esses dados positivos influenciam diretamente com a motivação dos profissionais envolvidos com a metodologia Extreme Programming (XP), pois através de sua simplicidade e práticas objetivas demonstraram a curto prazo sua eficiência no cenário do desenvolvimento de *software*, uma vez que possui um único objetivo: entregar projetos de forma célere, eficiente e de qualidade ao cliente.

Segundo Marchesi:

Extreme Programming (XP) é apoiado em práticas que formam a metodologia. A maioria das práticas XP também podem ser adotadas por desenvolvedores individualmente. Devido ao fato da metodologia XP ser muito bem detalhada e explicada, e não existir grandes dificuldades em seguir suas práticas, entretanto, é extremamente difícil e arriscado seguir todas as práticas rigorosamente e ao pé da letra de uma única vez (MARCHESI et al., 2002).

A grande vantagem da utilização da metodologia Extreme Programming (XP) dá-se ao seu ágil dinamismo, pois foi criada com o pensamento de inserção de mudanças,

onde é admissível realizar alterações e modificações dos requisitos no transcorrer do projeto, de acordo com a viabilidade e necessidade do cliente.

2.3.1 VALORES DO EXTREME PROGRAMMING (XP)

Extreme Programming (XP) como metodologia de desenvolvimento de *software* se baseia em quatro valores fundamentais:

- Comunicação;
- Coragem;
- *Feedback*;
- Simplicidade.

Comunicação: O intuito é diminuir eventuais falhas na comunicação de informações pertinentes ao projeto, substituindo, quando possível, o uso de documentações, e-mails e formalidades, visando a comunicação pessoal, eficaz e célere.

Ademais, é necessário que haja a comunicação entre os membros da equipe como um todo, diminuindo possíveis tomadas de decisões sem o prévio acompanhamento dos gestores do departamento, bem como alterações durante o desenvolvimento do projeto sem a participação e comunicação da equipe. Além disso, esta própria metodologia prevê que, quando houver a ausência de comunicação interna, um treinador irá reconhecer esta falha e integrar a comunicação novamente.

Simplicidade: A intenção é que cada membro da equipe possa expor suas ideias, soluções e opiniões, visando a facilitar a execução do projeto.

O objetivo de integrar a simplicidade aos trabalhos é justamente para que o projeto contenha códigos menos complexos e de fácil execução.

Segundo Beck:

Simplicidade e comunicação possuem uma maravilhosa relação de apoio mútuo. Quanto mais você comunica, mais claramente você é capaz de ver o que precisa ser feito e mais confiança você tem com o que realmente não precisa ser feito. Quanto mais simples é o seu sistema, menos você precisa comunicar sobre ele, o que leva à comunicação mais completa, especialmente se você for capaz de simplificar o sistema suficientemente a

ponto de necessitar de menos programadores (BECK, 2000, P. 31, tradução nossa).

Feedback: O *feedback* é imprescindível em todas as etapas do projeto, contudo, nesta fase ele irá possibilitar a melhoria contínua sobre todas as funcionalidades pertinentes ao projeto em questão.

Os desenvolvedores e analistas de sistemas obtêm *feedback* sobre a lógica e algoritmo da área de qualidade executando os casos de teste previamente construídos.

Já os clientes obtêm o *feedback* através dos testes funcionais realizados durante a etapa de homologação.

A psicologia do aprendizado ensina que o tempo entre uma ação e o correspondente *feedback* é crítico para o aprendizado. Experimentos com animais mostram que mesmo pequenas diferenças no tempo de *feedback* resultam em enormes diferenças de aprendizado. (...)

Portanto, um dos princípios é obter *feedback*, interpretá-lo, e colocar o que foi aprendido de volta dentro do sistema o mais rapidamente possível. As pessoas de negócio aprendem como o sistema pode contribuir da melhor forma e retornam este aprendizado em dias ou semanas, ao invés de meses ou anos. Os programadores aprendem como fazer o *design*, implementar e testar o sistema da melhor forma possível e retornam este aprendizado em segundos ou minutos ao invés de dias, semanas ou meses (BECK, 2000, p.37, tradução nossa).

Conforme Teles (2014, p. 51), para que o *feedback* exista em um projeto de *software*, é necessário que a comunicação esteja presente de forma muito intensa. A grande maioria dos sistemas é desenvolvida em equipes, de forma que, os componentes da equipe e o cliente precisam trocar ideias e informações para que o *software* ganhe forma e atinja os objetivos desejados.

Coragem: Durante o desenvolvimento de um *software*, podem ocorrer situações que podem gerar desespero aos integrantes do projeto. Esses desesperos possuem significativa influência, na maioria das vezes, pelos motivos listados abaixo:

- Alteração de código dos sistemas antigos;

- Exclusão do código para reescrever tudo de novo de forma simples;
- Compartilhamento do código com todos os envolvidos;
- Aceitação da opinião dos envolvidos e não tornar a sua o centro das atenções.
- Tempo insuficiente para a execução e entrega do projeto;
- Abrir mão da qualidade do produto desenvolvido em decorrência da falta de prazo hábil para executá-lo.

Estes temores são ocasionados pela ausência de confiança na utilização dos processos da metodologia que buscam, justamente, aumentar a proteção do projeto durante a fase de desenvolvimento.

2.3.2 PRINCÍPIOS DO EXTREME PROGRAMMING (XP)

Além dos valores supramencionados no tópico anterior, o Extreme Programming (XP) também possui um conjunto de princípios fundamentais para um bom desenvolvimento de *software*, sendo eles:

- *Feedback* rápido;
- Adotar simplicidade;
- Mudança incremental;
- Adotar mudanças, e
- Agregar qualidade.

O próprio Extreme Programming (XP) determina um conjunto de princípios que devem ser seguidos pelos membros da equipe do projeto, aqueles que forem necessariamente trabalhar com essa metodologia.

Sendo assim, apresentam-se a seguir os princípios que deverão auxiliar na solução de problemas durante a execução do projeto.

Feedback rápido: A equipe participante do projeto deve ter uma comunicação ativa durante o projeto, facilitando a resolução de qualquer problema ou, até mesmo, o aprendizado coletivo.

O *feedback* periódico é um princípio importante para o sucesso do projeto.

Adotar simplicidade: Os problemas devem ser tratados de forma simples no curso de todo projeto, pois podem ocorrer situações imprevisíveis e, quando isso acontece, a problemática deve ser tratada com simplicidade e resolvida da melhor forma possível.

Mudança incremental: O Extreme Programming (XP) propõe a realização de mudanças incrementais, que são alterações realizadas no projeto uma de cada vez. Este procedimento possui a cautela de evitar riscos e ameniza os efeitos das falhas. Sendo assim, os desenvolvedores podem realizar alterações de melhoria no código, algoritmos e, também, mudanças de requisitos. Porém, estas devem ser realizadas de forma incremental.

Dado que o sistema é desenvolvido de forma incremental, a equipe está continuamente fazendo a manutenção do *software* e criando novas funcionalidades. Em muitos casos, ela irá alterar algo que vinha funcionando corretamente, o que leva ao risco de gerar falhas no sistema. Por esta razão, a equipe precisa ser corajosa e acreditar que, utilizando as práticas e os valores do XP, será capaz de fazer o *software* evoluir com segurança e agilidade. (TELES, 2014, p. 26)

Adotar mudanças: A metodologia Extreme Programming (XP) busca facilitar a mudança de requisitos através dos princípios e práticas.

A ideia central é aceitar a mudança independentemente do nível em que se encontra o projeto. Inicialmente, pode parecer inviável, mas é benéfico quando aplicado aos projetos, pois quando ocorre uma grande mudança de requisitos, o cliente, muitas vezes, não consegue identificar o que ele realmente precisa, mas quando a ideia toma forma, ele consegue visualizar o sistema e verificar se está apto para atender suas reais necessidades.

Visto que os clientes normalmente não sabem direito o que desejam no início do projeto, eles tendem a pedir tudo o que acham que podem vir a precisar, especialmente se eles pensarem que só terão uma única chance de pedir. Esta é uma das melhores maneiras que conhecemos para aumentar o escopo do projeto muito além do necessário para alcançar a

missão geral do projeto (POPPENDIECK & POPPENDIECK, 2003, p.32-33, tradução nossa).

Agregar qualidade: Quando se utiliza uma metodologia ágil como o Extreme Programming (XP), a construção de um projeto deve priorizar a qualidade, onde esta não deve ser tratada como um critério opcional, mas sim, obrigatório.

Para o projeto atender a qualidade esperada, o sistema deve atender os requisitos de acordo com as necessidades do cliente, pois uma vez aprovado pela área de qualidade, haverá a homologação do sistema, e conseqüentemente, irá beneficiar não só o cliente pela qualidade e eficácia do produto, mas também, a empresa como um todo.

2.3.3 PRÁTICAS DO EXTREME PROGRAMMING (XP)

A metodologia Extreme Programming (XP) possui entre seus processos, doze práticas que constroem o núcleo principal da sua existência, que foram criadas utilizando como base os valores e princípios mencionados nessa monografia em tópicos anteriores.

Pode-se dizer, que as práticas utilizadas pelo Extreme Programming (XP) não são inovadoras, pois já eram utilizadas há muitos anos na área de *software* e trazem uma série resultados positivos.

Seguem, abaixo, algumas práticas da metodologia Extreme Programming (XP):

O jogo do planejamento: O planejamento é requisito essencial quando um projeto é iniciado.

Serão realizadas reuniões para que haja a definição dos requisitos que deverão ser entregues em cada “*release*” ou “*iteração*”. Neste momento, é imprescindível a colaboração de toda a equipe de desenvolvimento, inclusive o cliente, podendo haver a divisão das reuniões em duas partes:

- **Negócio:** Nesta reunião, deverão participar os membros que entendem sobre a regra do negócio da empresa, pois serão estabelecidos o que e quando o projeto poderá ser entregue, sem prejudicar o cronograma do cliente.

Para este encontro, não há a necessidade do comparecimento da equipe técnica do projeto.

- **Técnico:** Nesta reunião, deverão participar os membros que irão executar o projeto, ou seja, toda a equipe de desenvolvimento que necessariamente irá realizar a criação de código e algoritmos para o correto funcionamento do sistema.

Os desenvolvedores e analistas devem projetar e estimar o que será entregue em cada história e iteração de cada release.

No final de um *release*, é realizada uma reunião rápida para definir o escopo do próximo release, considerando as estimativas e cronograma do cliente.

Um *release* consiste de várias iterações e, em cada iteração, são compostas de várias histórias (requisitos simplificados). Os desenvolvedores devem estimar o cronograma de cada história e acordarem quantas histórias são possíveis realizar em cada *release*. E com essas informações e estatísticas, o cliente poderá escolher a ordem de cada entrega, conforme cronograma do produto.

Releases pequenos: Os *releases* devem ser compactos, o máximo possível, contudo, é evidente que eles devem possuir os requisitos mais importantes para o negócio.

Essa prática possibilita criar vários *releases*, resultando em *feedback*.

Metáfora: A intenção da metáfora é oferecer uma visão geral do sistema em um formato simples, que possa ser compartilhado por todos os envolvidos no projeto.

A ideia da metáfora é que seja realizada uma analogia entre o sistema que está sendo desenvolvido e outro sistema, não necessariamente de *software*, para que todos entendam, com o objetivo de obter um “vocabulário comum”.

Esta técnica visa interagir as áreas envolvidas, buscando o entendimento coletivo sem necessariamente entrar na parte técnica, em que é possível que os integrantes do projeto se expressem de uma forma simples e clara.

A metáfora também pode ser vista como uma forma simples de arquitetura do sistema, numa linguagem compreensível tanto para os clientes, como para os programadores.

Projeto simples: Pode-se explicar esta prática em duas partes:

A primeira é aquela em que devem ser projetadas as funcionalidades que já foram definidas, e não as que poderão ser alteradas futuramente.

A segunda é aquela em que deve ser elaborado o melhor projeto para ser entregue e atender todas as suas funcionalidades.

Estas práticas possuem o intuito de enfatizar que o projeto simples deve se concentrar em soluções também simples, contudo, bem elaboradas e estruturadas para o objetivo central do projeto.

As características do projeto simples são:

- Todos os testes realizados são executados com sucesso;
- O projeto deve expressar a ideia mais simples para ser desenvolvida;
- O projeto não deve possuir lógica duplicada;
- O projeto deve conter o menor número possível de classes e métodos.

Testes e Qualidade: Os testes na metodologia Extreme Programming (XP) são realizados antes da programação. Existem dois tipos de teste: teste de unidade e teste funcional.

Os **testes de unidade** são realizados para verificar os erros e falhas que possam existir antes do fim da construção do sistema e devem ser realizados antes do projeto ser encaminhado ao departamento de qualidade. Não é necessário escrever os testes de unidade para todos os métodos. Os testes de unidade são automatizados, e sempre que o programador escrever um novo código, ele irá verificar se o sistema passou em todos os testes.

Segundo Jeffries, as realizações de testes são imprescindíveis para o bom desenvolvimento do projeto:

No âmbito de Extreme Programming, qualquer recurso do programa, sem um teste automatizado, simplesmente não existe. Programadores devem escrever *unit tests* (toda a aplicação de teste nas assinaturas de entradas e saídas de um sistema, feito para validar dados via E/S) para que a sua

confiança no funcionamento do programa tornasse parte do próprio programa (JEFFRIES, 1999).

Os **testes funcionais** ou caixa preta são utilizados para verificação das funcionalidades desenvolvidas. Os testes servem para garantir a qualidade do sistema e identificar todos os requisitos planejados na fase inicial. Este mesmo teste é realizado pelo cliente na homologação do sistema.

Refatoramento: É a alteração no código ou algoritmo que são realizados visando melhorias no código, aumentando sua capacidade de adaptação às mudanças ou, até mesmo, para melhorar a sua qualidade.

O refatoramento deve sempre aprimorar o código já existente, tornando-o mais simples e melhor estruturado, sem alterar sua funcionalidade.

A refatoração mantém a semântica do código, ou seja, após as mudanças, o código ainda funciona da mesma forma. Por exemplo, se o nome de uma operação de uma classe for mudado, todo o código do seu sistema que invoca aquela operação passará a referenciar o novo nome, e não o antigo (FOWLER, 2004).

Programação em pares: Uma Prática muito importante na metodologia Extreme Programming (XP), pois é escrito por dois programadores, que possuem funções distintas.

A formação das duplas é dinâmica. Se duas pessoas pareiam pela manhã, à tarde eles poderiam facilmente trabalhar com outras pessoas. Se um dos membros do time tem a responsabilidade de uma tarefa em uma área que lhe é estranha, pode pedir a alguém com experiência nessa área para parrear. É uma forma de passar conhecimento para todos no time (BECK, 1999).

Estes programadores devem sempre estar sentados um ao lado do outro, olhando para o computador, focadamente, no momento da construção do código propriamente dito.

Sendo assim, um programador será responsável pela codificação, algoritmos e lógica de programação. E o segundo programador observará o código produzido,

com um olhar mais estratégico buscando melhorá-lo e torná-lo mais simples, além de apontar possíveis erros e pontos falhos. Além do mais, também serve como *backup* caso o programador por alguma inviabilidade pessoal, não possa participar do projeto.

Ademais, estas duplas são constantemente alteradas, juntamente com os papéis que os desenvolvedores realizavam, com o intuito de diminuir eventual margem de erro, e também, para que todos os membros da equipe possam estar atualizados com o *status* do sistema e de seu desenvolvimento.

Propriedade coletiva do código: A codificação e estrutura da lógica é realizada por dois programadores.

A princípio, pode-se pensar que esta prática não é eficaz, contudo, é preciso coragem e ousadia para implantar essa prática, pois todos devem respeitar o mesmo padrão de codificação, realizando todos os testes para verificação de erros.

Este método é benéfico e ajuda a equipe a trabalhar em sintonia, procurando atingir um melhor resultado e qualidade para o projeto desenvolvido. Esta atividade deve ser realizada de uma forma controlada e pode ser facilitada com o uso de uma ferramenta de controle de versão.

Integração contínua: Uma integração contínua deve ser adequada dentro da equipe.

A integração do código deve ser integrada sempre que houver a finalização de uma funcionalidade.

A empresa pode buscar a melhor forma de realizar essa prática, contudo, o modo mais simples se dá com a disponibilização uma máquina dedicada somente para integração.

Os dois programadores com o seu código, devem ocupar a máquina, carregando a versão corrente do sistema, as alterações que promoveram, e após isso, proceder com a realização dos testes de unidade até que eles sejam aprovados em sua integralidade.

Um ponto crucial na integração é que só poderá haver a integração das funcionalidades, quando naquelas inseridas não houver erros, caso contrário, estes devem ser corrigidos para então serem integrados.

Semana de quarenta horas: Uma das práticas muito difundidas no Extreme Programming (XP) é a semana de quarentas horas.

Esta semana obriga as equipes de projetos a trabalharem apenas 40 (quarenta) horas por semana.

Segundo Beck (1999), essa prática procura ratificar o foco nas pessoas e não em processos e planejamentos. Caso seja necessário, os planos devem ser alterados, ao invés de sobrecarregar as pessoas.

Esta é uma filosofia importante, pois uma empresa que não possui uma metodologia definida, está fadada a realizar demasiadas horas extraordinárias e, com isso, as equipes acabam sobrecarregadas, ocasionando insatisfação no ambiente de trabalho e, resultando numa queda de qualidade do código que está sendo desenvolvido.

Cliente alocado: O cliente sempre deve estar presente no transcorrer do projeto, devendo, literalmente, fazer parte da equipe de desenvolvimento, pois terá a tarefa de auxiliá-la sanando questionamentos e possíveis dúvidas. Mesmo que não haja a possibilidade de se ter o cliente alocado na empresa, deve-se ter alguém que, no mínimo, tenha o conhecimento suficiente do negócio para exercer esta função. Quando isso ocorre, geralmente, quem realiza esse papel é o analista de negócio, porém, vale ressaltar ainda, que o importante e ideal é que o cliente esteja presente.

Padrões de codificação: Como Extreme Programming (XP) prega a propriedade coletiva de código, onde há a possibilidade de todos poderem fazer alterações e realizarem o refatoramento de qualquer parte do código, a qualquer momento, é mais do que necessário que haja padrões específicos de codificação.

O objetivo é que todos os envolvidos programem da mesma forma, facilitando o entendimento do código, suas alterações, visando à criação de um código limpo.

As vantagens dos padrões de código é que eles representam o estilo de programação de toda a equipe de desenvolvimento, por esta razão utilizar os padrões de código agilizam a programação economizando tempo se compararmos com o tempo gasto para a compreensão de um código sem endentação e nomes de variáveis sem significado algum para o contexto do programa (BECK, 1999).

2.3.4 CICLO DE VIDA DE UM PROJETO EXTREME PROGRAMMING

Os projetos que utilizam a metodologia Extreme Programming (XP) passam por algumas fases durante o seu ciclo de vida. Essas fases são compostas de várias tarefas que devem ser necessariamente executadas.

Sendo assim, a seguir serão demonstrados, brevemente, com uma explicação sobre o ciclo de vida de um projeto, juntamente com suas principais fases, com o intuito de se adquirir uma superficial noção do projeto, desde o início até o seu fim.

- Exploração;
- Planejamento inicial;
- Iterações do *release*;
- Produção;
- Manutenção;
- Morte.

Fase de exploração: Essa fase é a inicial para a construção do sistema.

Nela, serão analisadas as possíveis ideias, verificando a viabilidade das mesmas e prováveis soluções.

Os programadores elaboram arquiteturas dos sistemas e apresentam como o este funcionará, considerando o ambiente tecnológico e técnico.

Com isso, os programadores podem apresentar uma ideia do que pode ser feito com os requisitos discutidos e quando eles possuírem histórias suficientes, já poderão iniciar a construção do primeiro *release* do sistema.

Recomenda-se que seja gasto, no máximo, duas semanas nessa fase inicial.

Fase de planejamento inicial: Essa fase é definida quando o cliente concorda como o primeiro *release*.

O planejamento funciona da seguinte forma: os programadores, juntamente com o cliente, definem as histórias que serão implementadas e as descrevem em cartões.

Após, os programadores assinalam o grau de dificuldade para a execução de cada história, e baseados na velocidade de implementação, informam quantas histórias podem implementar em uma iteração. Ato contínuo, os clientes escolhem as histórias de maior valor para ser implementadas na iteração.

Sendo assim, este processo se repete até finalizar todas as iterações do *release*. O tempo para cada iteração deve ser de uma a três semanas e para cada *release*, de dois a quatro meses.

Fase das iterações do release: Aqui, serão escritos os casos de testes funcionais e de unidade.

Os programadores irão seguir o seguinte fluxo de atividades nesta ordem (em cada iteração): escrita dos casos de testes; projeto e refatoramento; codificação; realização dos testes; e integração.

À medida que esse fluxo evolui, o sistema, aos poucos, vai sendo construído segundo os princípios, valores e práticas apresentadas nesta monografia.

Após o término do primeiro *release*, já é possível se ter uma noção das tecnologias e domínio do problema, de modo que as iterações poderão ser mais curtas nos *releases* subsequentes, o que já é possível fazer estimativas com um nível maior de confiabilidade, baseando-se nas iterações passadas.

Fase de produção: Inicia-se após a construção do sistema, que depois de construído, é colocado para rodar em um ambiente que simula o ambiente de produção, para que seja possível ver o seu comportamento e desempenho em termos de performance.

Fase de manutenção: Pode-se considerar como uma característica inerente a um projeto que usa a metodologia Extreme Programming (XP).

Em projetos em que há a produção de novas funcionalidades, mantendo o sistema existente em funcionamento, pois no decorrer do projeto podem entrar novos integrantes à equipe, podendo existir outras melhorias no código.

Nesta fase de manutenção, deve-se usar os seguintes mecanismos:

- Refatoramento;
- Introdução de novas tecnologias;
- Introdução de novas ideias de arquitetura.

É importante ressaltar que a manutenção dada em um sistema que já está em andamento deve ser realizada com excesso de cautela, pois uma alteração

incoerente pode paralisar o funcionamento do sistema, resultando em prejuízos para o projeto, para o cliente, e também, para a empresa.

Fase de morte: Esta fase é a que corresponde ao término do projeto.

Existem duas razões para se chegar ao final de um projeto: uma boa e a outra ruim. A razão boa se dá quando o cliente se encontra satisfeito com o sistema existente e não enxerga nenhuma funcionalidade que possa vir a ser implementada no futuro. Já a má razão dada para a morte em Extreme Programming (XP) é ocasionada quando o projeto se torna economicamente inviável, devido às dificuldades encontradas para adicionar funcionalidades, a um custo baixo e também, pela existência de erros reiterados.

2.3.5 PAPÉIS DENTRO DO PROJETO USANDO EXTREME PROGRAMMING (XP)

A gerência do projeto tem o principal papel de tomada de decisões que, não podem ser tomadas por uma equipe.

Contudo, é necessário que haja a descentralização das pessoas responsáveis aptas a autorizar ou não algo, e para que isso ocorra de forma organizada, é imprescindível que haja o andamento do jogo do planejamento: coletar métricas e fazer com que elas sejam vistas por àqueles, cujo o trabalho esteja sendo medido, bem como, intervir em situações de problemas.

A gerência deve trabalhar exaustivamente com as métricas, pois precisa verificar constantemente se o tempo estimado para desenvolver o projeto está de acordo com o tempo que realmente é gasto. Durante o desenvolvimento do projeto, há a cristalina possibilidade de se auferir o tempo que está sendo despendido para a elaboração do projeto, com aquele estimado.

Ademais, a gerência também deverá verificar se a equipe está trabalhando corretamente e evoluindo com o projeto como um todo e caso se não esteja, será necessário que haja a intervenção de maneira corretiva, para a solução do problema.

A melhor forma de disponibilizar isso para os membros da equipe é através da exposição das métricas, em forma de gráfico atualizado – se possível, semanalmente.

A metodologia Extreme Programming (XP) utiliza a seguinte divisão: o treinador (*coach*) e o rastreador (*tracker*). Esses papéis podem ou não ser executados pela mesma pessoa.

- O treinador se preocupa com a execução técnica do projeto, essa tarefa é importante para o andamento do mesmo;

Deve ser um bom comunicador e possuir um bom conhecimento técnico, mantendo a calma para realizar tarefas e solucionar problemas sob pressão;

O papel do treinador não é de tomar decisões técnicas, mas sim, de fazer com que todos os membros da equipe escolham as melhores decisões para ser seguidas e também, de facilitar o processo de desenvolvimento.

- O rastreador se preocupa em realizar métricas sobre o que está sendo desenvolvido, confrontando com as já estimadas e verificando possíveis divergências.

Uma forma recomendada para tal tarefa se dá através da coleta de dados duas vezes por semana.

Além dos papéis gerenciais apresentados anteriormente, uma equipe que utiliza Extreme Programming (XP) para desenvolver *software* é composta de outros papéis, sendo eles: desenvolvedor, cliente, *quality assurance* e consultor.

Desenvolvedor: é o profissional especializado no desenvolvimento e aperfeiçoamento de programas, sendo responsável para realizar importantes tarefas no transcurso do projeto, cabendo a ele:

- Analisar;
- Projetar;
- Testar;
- Codificar;
- Integrar o sistema.

O programador também realiza a estimativa e ordena as criticidades das histórias. Na metodologia Extreme Programming (XP) o foco está em entregar funcionalidades

implementadas para o cliente. Ele está constantemente escrevendo testes de unidade, codificando e fazendo refatoramento com o objetivo de produzir um código de alta qualidade e com celeridade.

Cliente: É aquele responsável pela escolha do produto final. Ele determina o que é prioridade ou o que pode ser adiado. Além disso, o cliente define, com a ajuda dos testadores, os testes funcionais que irão demonstrar se o sistema elaborado realmente satisfaz as suas necessidades ou se precisa de alterações e/ou mudanças.

Quality Assurance: Possui papel de extrema relevância no percurso do projeto. Este profissional garante a qualidade do produto, pois possui elevado nível de conhecimento acerca das regras do negócio. Suas principais atribuições são:

- Analisar;
- Planejar os testes que serão executados no projeto;
- Executar todos os testes;
- Identificar “não-conformidades”;
- Verificar a ausência de algum requisito solicitado pelo cliente;
- Auxiliar o desenvolvedor na criação dos testes de unidade.

Além do mais, é responsável pelo suporte ao cliente na realização dos testes no ambiente de homologação.

Consultor: Para ser detentor deste cargo, que é necessário somente em algumas situações específicas, é exigido a presença de algum profissional com elevado nível de conhecimento e *expertise*, em uma determinada tecnologia, sobre determinado assunto, o qual não está sendo bem dominado pelas pessoas que participam do projeto. Geralmente, um profissional relacionado com a área em que o projeto será utilizado.

3. ESTUDO DE CASO

3.1. INTRODUÇÃO

Terceira colocada no mercado de *softwares*, a empresa que aqui será estudada foi criada em meados de 2011. Localizada na região de São Paulo, no centro, possui mais de dois mil funcionários alocados no edifício. E, para esta monografia, será utilizado o nome fantasia de "Certified Inovation S.A.", para promover o sigilo da empresa em apreço.

No ano de 2012, ocorreu uma crise nos projetos desenvolvidos pela empresa, uma vez que seus gastos eram superiores ao valor do projeto inicial, o que ocasionou em perda de capital intelectual, perda de grandes clientes com potencial, de mão de obra qualificada, existência de gastos elevados com horas extraordinárias, pagamento de multas e a perda de credibilidade no mercado em que atua.

O faturamento da empresa era de aproximadamente R\$ 2 milhões no ano de 2012, decaindo drasticamente, uma vez que receita da empresa estava diretamente ligada com a venda dos projetos de *softwares*.

O presente estudo de caso possui como objetivo demonstrar que é possível reverter um quadro de declínios – ocasionado pela ausência de uma boa gestão de processos –, através da utilização de uma metodologia moldada aos costumes da empresa, promovendo maior organização, qualidade e lucratividade.

3.2. METODOLOGIA UTILIZADA

Após apurar os problemas internos da empresa, a equipe de pesquisa estudou as melhores metodologias existentes no mercado e, com o consenso de todos os envolvidos, chegou-se à conclusão que poderia solucionar as falhas encontradas através da inserção de uma nova metodologia criada de acordo com a cultura da empresa.

Essa metodologia eleita foi desenvolvida e baseada em três outras metodologias já existentes no mercado, sendo elas: PMBOK, SCRUM e Extreme Programming.

Com base nestas três metodologias, a equipe de pesquisa apresentou uma nova metodologia criada a partir daquelas pesquisadas.

Inicialmente, antes da criação de uma nova metodologia, foi definido que a sua implantação, dar-se-ia paulatinamente, ou seja, a inserção desta seria realizada aos

poucos, por cautela – para evitar possíveis prejuízos a empresa – e também, para o devido aprimoramento e aperfeiçoamento de suas funcionalidades, de acordo com as necessidades específicas do departamento em questão.

Sendo assim, ela foi implementada somente no departamento de desenvolvimento, a qual possuía uma grande demanda de trabalho, e também, onde seria mais fácil identificar se a nova metodologia estaria atingindo os objetivos, sendo mais eficaz, e passível de testes para aperfeiçoar suas funções e funcionalidades.

Os princípios utilizados para a criação da nova metodologia, que foi implantada posteriormente na empresa, deram-se em decorrência do embasamento de diversos valores e procedimentos importantes para a efetivação e inserção da “cultura de processos”.

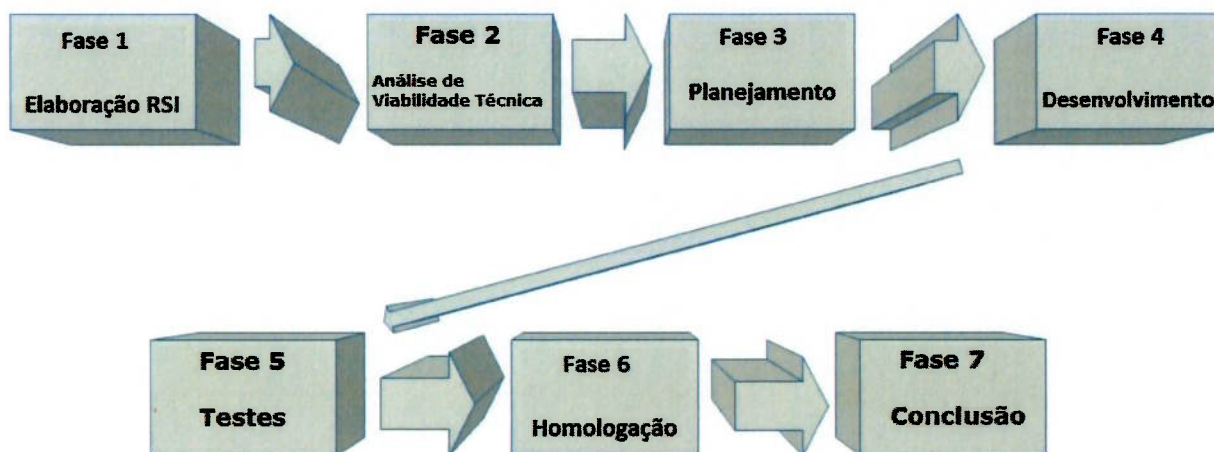
Para tanto, seguem abaixo as principais funcionalidades utilizadas para a criação e implantação da nova metodologia.

- Para o período que se estendeu da ideia inicial ao começo da comercialização de um produto, foi estabelecida uma metodologia funcional para que pudesse atuar em diversas áreas da Certified Inovation S.A, determinando as diretrizes para o acompanhamento do projeto;
- Foi necessário que houvesse a uniformização de todos os procedimentos com o fito de aprimorar os planejamentos, controlar e acompanhar as atividades envolvidas no desenvolvimento de projetos;
- Com a implantação da nova metodologia foi possível aumentar a qualidade e a confiabilidade das soluções;
- Com uma nova gestão de processos e com a utilização de um meio eficaz, houve considerável aumento na produtividade dos colaboradores da empresa estudada - a Certified Inovation S.A;
- Foi possível potencializar a visibilidade dos projetos aos usuários, áreas de operação, produto suporte, gerentes e desenvolvedores;
- Com a criação de uma base histórica de métricas e indicadores elaborados a partir dos projetos realizados, houve a possibilidade de se estabelecer parâmetros com precisão e confiabilidade nos orçamentos e cronogramas.

3.2.1. DESENVOLVIMENTO DE PROJETO

Quando um novo produto (interno ou requisitado por algum cliente) torna-se de interesse para a empresa, uma equipe é formada para a sua definição e elaboração. O processo de desenvolvimento deste novo produto é dividido em diversas fases distintas, que conforme o desenrolar do projeto, se complementam, uma vez que o mesmo só poderá avançar de acordo com a conclusão da fase anterior, segundo o esquema supra.

Figura 1 - Fases do desenvolvimento da nova metodologia



3.2.2. TAMANHO DOS PROJETOS

Para que a equipe de desenvolvimento possa ter a percepção do tamanho do projeto, é necessário que haja uma estruturação básica de horas para ter uma noção da complexidade do projeto no que diz respeito à sua estrutura, sendo:

- Projeto de tamanho grande (superior a 600 horas)
- Projeto de tamanho médio (de 200 horas até 600 horas)
- Projeto de tamanho pequeno (até 200 horas)

Para se determinar o tamanho do projeto é preciso identificar a sua estrutura em requisitos e realizar uma análise técnica.

A metodologia implantada no setor de desenvolvimento, foi criada através de algumas fases, que para facilitar o entendimento e visualização do novo método – aqui, serão demonstradas através de quadros, contendo sua estrutura, processos utilizados, atividades que serão desenvolvidas, prazo que deverão ser cumpridos, documentação necessária, bem como, o colaborador que será responsável para proceder com a finalização da fase.

Para iniciar um projeto, é necessário que já tenha sido realizado, previamente, o RSI – Relatório de Solicitação Inicial, que consiste na elaboração de um documento, preenchido pelo cliente – interno ou externo.

Quadro 1 - RSI – Relatório de Solicitação Inicial

Elaboração RSI	Critério de Início: Elaboração do RSI (Relatório de Solicitação Inicial)
Análise de Viabilidade Técnica	Objetivo: Caracterizar de forma íntegra a solicitação, efetuar o levantamento dos requisitos funcionais e quaisquer outras informações necessárias para entendimento do escopo da solicitação. Atividades: • Elaboração do documento RSI: - Premissas; - Características; - Descrição da situação atual; - Descrição das necessidades / problemas; - Expectativas de prazo; - Mercado a ser atingido; - Definição do cliente, interessados e aprovadores. Documentos: • RSI (Relatório de Solicitação Inicial); • ATA da reunião de entendimento. Responsável pela finalização: Operações / Produtos / PKI / Desenvolvimento Aprovadores: Não há Critério de Fim: RSI finalizado e reunião de entendimento. Observação: Durante a fase de RSI não há alterações para projetos de tamanho grande, médio e pequeno.
Planejamento	
Desenvolvimento	
Testes	
Homologação	
Conclusão	

Este relatório facilita a entrada do projeto, pois ele é composto pelos requisitos que o produto deverá conter, e também com as regras de negócio – ainda que parciais.

Nesta fase, a equipe de desenvolvimento não possui nenhuma interferência externa, uma vez que a elaboração do documento é realizada pelo próprio cliente, e somente após a entrega deste documento é possível iniciar o desenvolvimento do projeto.

O RSI é uma visão inicial e superficial dos requisitos funcionais e quando é finalizado e entregue à área de desenvolvimento, é possível se realizar uma breve análise e, em seguida é designada a primeira reunião.

Na reunião, serão discutidos, de maneira específica, todos os requisitos e as regras do negócio. Poderão ser realizadas reuniões futuras, de acordo com a complexidade do projeto.

Sendo assim, a próxima fase se iniciará.

Para a construção da AVT – Análise de Viabilidade Técnica, a equipe de desenvolvimento realiza o *Product Backlog*, que deverá conter uma lista com as funcionalidades desejadas e com todas as informações anteriormente documentadas.

Quadro 2 - AVT – Análise de Viabilidade Técnica

Elaboração RSI	Critério de Início: RSI (Relatório de Solicitação Inicial) disponível para análise.
Análise de Viabilidade Técnica	
Planejamento	Objetivo: Desenvolver o estudo da viabilidade técnica do produto, devendo basear-se em uma análise da solicitação e avaliação dos recursos, atividades e prazos estimados para o produto/projeto.
Desenvolvimento	
Testes	
Homologação	
Conclusão	
	Atividades: - Execução da Análise de Viabilidade Técnica: - Detalhamento do escopo da solicitação; - Verificação de soluções semelhantes na empresa (opcional); - Levantamento e análise de alternativas (opcional); - Descrição das interfaces (opcional); - Solicitação de requisitos funcionais e não funcionais; - Determinação das premissas, restrições e exclusões específicas do produto/projeto; - Determinação das funcionalidades do produto/projeto; - Levantamento das necessidades de recursos (Humanos Hardware e Software) (opcional); - Identificação e análise dos riscos iniciais do projeto;

(Continua)

	- Estimativa macro de Tamanho, Esforço, Prazo.
	Documentos: • AVT (Análise de Viabilidade Técnica); • ATA da reunião de aprovação da AVT;
	Responsável pela finalização: Gerente / Coordenador desenvolvimento.
	Aprovadores: serão definidos na RSI (cliente) e gerente desenvolvimento.
Critério de Fim: Reunião de aprovação do documento AVT.	

Na AVT, com todas essas informações devidamente documentadas, é possível se ter a rastreabilidade das informações para eventuais consultas.

Após a conclusão do documento pela equipe de desenvolvimento, será marcada uma reunião para a apresentação das funcionalidades, fluxo de telas (*wireframes*), bem como a estimativa macro de tamanho, esforço e prazo.

Esta reunião possui como objetivo inicial, a aprovação do documento, para que então, a equipe de desenvolvimento possa prosseguir para a fase seguinte da metodologia.

A seguir, será demonstrado a realização do planejamento do projeto, sendo uma das partes mais importantes da metodologia, pois, com a presença de alguma uma falha no planejamento, poderá ocasionar um risco futuro para a entrega do projeto. Por este motivo, é imprescindível que a fase de planejamento deve ser realizada com o auxílio de todos membros envolvidos no projeto.

Quadro 3 - Planejamento do Projeto

Elaboração RSI	Critério de Início: AVT (Análise de Viabilidade Técnica) disponibilizada e aprovada.
Análise de Viabilidade Técnica	
Planejamento	Objetivo: Criar o plano do projeto para execução e acompanhamento do mesmo. É preciso que haja a preparação uma definição completa e detalhada das funções lógicas e inter-relacionamentos entre sistemas/serviços.
Desenvolvimento	
Testes	
Homologação	
Conclusão	
	Atividades: • Planejamento do Projeto: - Detalhamento de requisitos (opcional pequeno); - Estruturação e formalização da equipe do projeto;

(Continua)

	- Elaboração do cronograma; - Estimativas detalhadas de Tamanho, Esforço, Prazo (<i>Backlog do Sprint</i>); - Entrega das histórias em cada (<i>Sprint</i>); - Estabelecimento dos compromissos internos e externos do projeto; - Definição da estratégia de Monitoramento e Controle; - Elaboração do Plano de Comunicação, incluindo o planejamento das Reuniões de acompanhamento de projeto (<i>Daily Scrum</i>); - Documentação Técnica (descrição detalhada de processamento físico dos fluxos de informação) (opcional pequeno); - o Definição dos <i>layouts</i>; - o Especificação dos componentes; - o Especificação Caso de Uso / Fluxogramas; - o Especificação Fluxo de Telas. - Criação de atividades 'macro' no quadro de tarefas (<i>Kanban</i>); - Criação atividades no <i>Redmine</i>; - Atualização atividades / horas gastas no <i>Redmine</i>; Documentos: • Plano do Projeto; • Documentação Técnica; • ATA da reunião de <i>kick off</i>. Responsável pela finalização: Gerente / Coordenador desenvolvimento. Aprovadores: Não há Critério de Fim: Planejamento do Projeto finalizado e realização da reunião de <i>kick off</i>.
--	--

No planejamento, serão devidamente projetadas as tarefas que todas as equipes envolvidas deveram executar no decorrer do projeto.

Será realizado um cronograma baseado na estimativa de cada membro através do *Backlog do Sprint*, com isso ocorre a definição da estimativa, esforço que será desempenhado e o prazo para o cumprimento do projeto. Sendo assim, a equipe de desenvolvimento divide as histórias de cada *Sprint*, e com a finalização da reunião, a equipe já possui o conhecimento de todos os procedimentos que ainda serão realizados, bem como a divisão para entregas das funcionalidades ao cliente.

Ato contínuo, após uma visão geral das entregas, a equipe iniciará com a fase 4 do projeto, conforme o quadro a seguir.

Quadro 4 - Desenvolvimento do Projeto

Elaboração RSI	Critério de Início: Planejamento finalizado. Objetivo: Esta é a fase de execução do projeto, onde os produtos definidos no documento técnico serão desenvolvidos detalhadamente e testados, ainda no ambiente de desenvolvimento. Atividades: • Codificação dos programas / Desenvolvimento do Produto; • Elaboração das instruções de instalação e produção/operação; • Criação do código e algoritmo em pares (Dois programadores); • <i>Daily Scrum</i> (Diário); • Criação da tabela <i>Kanban</i> e divisão das histórias (XP/SCRUM); • Execução de testes de unidade (opcional); • Revisão de prazo e esforço; • Revisão dos planos de projeto; • Execução do monitoramento e controle do Projeto; • Realização de Reuniões de Acompanhamento de Projeto; • Criação casos de teste (<i>Testlink</i>); • Atualização atividades / Horas gastas no <i>Redmine</i>; • Acompanhamento do cliente; • Criação de <i>Release Note</i> para ambiente de QA. Documentos: • Conjunto de programas desenvolvidos e documentados; • Manual de Instalação; • Casos de teste no <i>Testlink</i>; • <i>Release Note</i>. Responsável pela finalização: Gerente / Coordenador desenvolvimento Aprovadores: Não há Critério de Fim: Testes realizados pela equipe de QA no ambiente de desenvolvimento.
Análise de Viabilidade Técnica	
Planejamento	
Desenvolvimento	
Testes	
Homologação	
Conclusão	

O desenvolvimento do projeto em si é a própria execução do projeto, através da nova metodologia.

No quadro 4 houve a combinação de três metodologias, com o intuito de potencializar uma fluidez melhor na execução do projeto, sendo possível desenvolver com mais qualidade o produto final.

É possível observar também que a codificação é realizada em pares, ou seja, com dois programadores trabalhando simultaneamente, utilizando códigos simples e limpos para desenvolver o *software*.

Nesta fase, houve também a realização diária do *Daily Scrum*, onde a definição do horário do *Daily* fica a critério da equipe de planejamento, sendo recomendável que ela seja realizada no período da manhã, com a presença de todos os membros, inclusive o cliente. Aqui a comunicação é mais interativa, usando a metodologia do SCRUM.

Importante salientar que o *Daily* é um importante instrumento para saber o real andamento do projeto, sendo possível detectar eventuais falhas que possam ocasionar atrasos para a conclusão do mesmo e o fiel cumprimento do cronograma.

Na fase de desenvolvimento, também é realizada uma apresentação das funcionalidades já criadas, assim, caso exista alguma modificação a ser realizada, esta poderá ser alterada ainda na etapa de desenvolvimento.

Ademais, para a realização do *Release/Sprint* o cliente é chamado para uma apresentação prévia daquela funcionalidade em particular. Este procedimento é realizado até a construção final projeto ou quando o sistema desenvolvido já está completo.

Com o termino do desenvolvimento a equipe de qualidade pode atuar, iniciando a próxima fase.

Quadro 5 - Teste e Qualidade do Produto

Elaboração RSI	Critério de Início: Solução disponível para execução dos testes sistêmicos e de integração em ambiente de Desenvolvimento. Objetivo: Testar o sistema como um todo, suas interfaces com outros sistemas, validando padrões e segurança de informações, em ambiente de teste. Atividades: • Preparação de recursos para testes (programas, procedimentos, base de informações e instruções de instalação); • Preparação do ambiente de testes;
Análise de Viabilidade Técnica	
Planejamento	
Desenvolvimento	
Testes	
Homologação	
Conclusão	

(Continua)

	V • Execução dos testes e atualização <i>bugzilla</i>; • Verificação da integração com outros sistemas; • Correção de <i>BUGS</i> / Verificações e atualizações <i>bugzilla</i>; • <i>Daily Scrum</i> (Diário); • Atualização do quadro <i>Kanban</i> das histórias (XP/SCRUM); • Revisão de prazo e esforço; • Revisão do Plano de projeto; • Execução do monitoramento e controle do Projeto; • Realização de Reuniões de Acompanhamento de Projeto; • Criação de GMUD para ambiente de homologação; • Criação de <i>Release Note</i> para ambiente de homologação; • Atualização atividades / horas gastas no <i>Redmine</i>; • Criação de manuais (técnico e usuário); Documentos: • Conjunto de programas testados e documentados; • Manuais (técnico e de usuário). • GMUD • <i>Release Note</i> Responsável pela finalização: Gerente de Desenvolvimento / Coordenador de QA Aprovadores: Não há Critério de Fim: Aprovação dos testes e sistema liberado em ambiente de QA.
--	---

A qualidade é de extrema importância para os projetos, e com isso, nesta fase, serão realizados diversos testes para identificar se os requisitos foram entregues conforme as instruções do cliente, e se os mesmos atingem a finalidade em questão. Será observado se o *software* está funcionando corretamente e em sua integralidade.

É de responsabilidade da área de qualidade, realizar testes com base nos requisitos documentados, através da ferramenta *open source*, chamado de *Testlink*, para que, então, os casos de testes permaneçam na ferramenta.

Este processo é realizado somente quando o pacote é liberado para o ambiente de testes, no qual a equipe de qualidade irá proceder com os testes com base no planejamento.

Após testar os requisitos e funcionalidades na execução do teste, se a equipe de qualidade encontrar alguma não-conformidade, esta deverá reportar a mesma utilizando outra ferramenta: *open source*, também chamada de *Bugzilla*. Através

desta ferramenta, é possível realizar o reporte do erro, identificando o cenário que ocorreu a falha.

Sendo assim, após o analista de teste reconhecer e documentar a falha na ferramenta *Bugzilla*, esta ferramenta envia um *e-mail* para todos os envolvidos no projeto e, enquanto a equipe de qualidade executa os testes, a equipe de desenvolvimento corrige as não-conformidades identificadas.

Neste período, a equipe de qualidade pode realizar várias baterias de testes, até o pacote ou o sistema tornar-se estável, atingindo a qualidade pretendida.

Após a finalização dos testes, o sistema é liberado para o ambiente de homologação. Nesta fase, a equipe de desenvolvimento não atua em sua integralidade, mas ainda presta alguns suportes até a efetiva aprovação para o ambiente de produção.

No quadro abaixo, segue a de homologação do sistema.

Quadro 6 - Homologação do Projeto

Elaboração RSI	Critério de Início: Execução dos Testes de Aceitação e Sistema liberado em ambiente de QA.
Análise de Viabilidade Técnica	
Planejamento	Objetivo: Aprovar o sistema pelo usuário, validando requisitos, padrões e critérios de segurança de dados em ambiente de aceite.
Desenvolvimento	
Testes	Atividades: • Preparação do ambiente de aceite; • Preparação dos recursos para aceite; • Realização de treinamento; • Acompanhamento dos testes do cliente pela equipe de QA; • Solicitações de mudança para correção de problemas detectados; • Execução de correções de problemas detectados; • Homologação do Sistema; • Testes do <i>Check-List</i>; • Revisão dos planos de projeto; • Realização de Reuniões para alinhamento com o cliente; • Atualização atividades / horas gastas no <i>Redmine</i>.
Homologação	
Conclusão	

(Continua)

	Documentos:
	• Plano de Homologação; • Plano de Testes para homologação; • Atas de Reuniões; • Documento de aprovação.
	Responsável pela finalização: Gerente de desenvolvimento / Coordenador de desenvolvimento / Coordenador de QA
	Aprovadores: Não há
Critério de Fim: Sistema Homologado / Documento de Aprovação.	

Na fase de homologação, o sistema é homologado pelo cliente, ou seja, o próprio cliente valida o projeto realizado através de uma vasta e minuciosa pesquisa em todos os requisitos.

Antes disso, a equipe de qualidade realiza uma apresentação do sistema, e também, o devido treinamento ao cliente acerca as funcionalidades do sistema e uso do mesmo.

Após o aceite do cliente, a fase de homologação se encerra e o sistema será encaminhado para o ambiente de produção.

Quadro 7 - Conclusão do Projeto

Elaboração RSI	Critério de Início: Sistema Homologado.	
Análise de Viabilidade Técnica		
Planejamento		
Desenvolvimento		
Testes		
Homologação		
Conclusão		
	Objetivo:	
	Liberar o sistema para plena utilização, acompanhando o processo de implantação de forma a garantir o seu funcionamento no período inicial, em ambiente de produção.	
	Atividades:	
	• Criação de GMUD para ambiente de produção; • Criação de Release Note para ambiente de produção; • Atualização atividades / horas gastas no Redmine; • Apresentação do desenvolvimento realizado a equipe.	
	Documentos:	
	• GMUD • Release Note	
V	Responsável pela finalização: Operações / Produtos / PKI	
	Aprovadores: Não há	
Critério de Fim: Sistema Implantado / Produto pronto para produção.		

Na fase de conclusão do projeto, a equipe de desenvolvimento não terá mais participação após a aprovação do sistema em produção. Os procedimentos de instalação e configuração serão realizados pela equipe de infraestrutura, produtos e operações.

Panorama Geral do Desenvolvimento do Projeto

Na tabela abaixo, foram devidamente e minuciosamente descritos todos os procedimentos que são realizados durante cada fase do desenvolvimento do projeto, para que seja possível visualizar conjuntamente a evolução das tarefas executadas em cada fase.

Quadro 8 – Gerenciamento de atividades

FASE	PRAZO	ATIVIDADE	DOCUMENTO	RESPONSÁVEL
RSI	No máximo 5 dias úteis	Elaboração do documento RSI	RSI	Cliente (Operações / Produtos / PKI)
RSI	1 dia	Reunião	Ata de reunião	Desenvolvedores QA Coordenadores
AVT	Período de finalização, contabilizando a criação do documento e a realização da reunião de aprovação do: Projeto Pequeno: 3 dias Projeto Médio: 5 dias Projeto Grande: 11 dias	Execução da Análise de Viabilidade Técnica	AVT	Desenvolvedores QA Coordenadores
AVT	1 dia	Reunião	Ata de reunião	Desenvolvedores QA Coordenadores
Planejamento	Determinado conforme planilha de esforço	Detalhamento de requisitos	Documento de Requisitos	QA
Planejamento	Determinado conforme planilha de esforço	Estruturação e formalização da equipe do projeto	Plano do projeto	Coordenador Desenvolvimento
Planejamento	Determinado conforme planilha de esforço	Elaboração do Cronograma	Plano do projeto	Coordenador Desenvolvimento
Planejamento	Determinado conforme planilha de esforço	Estimativas detalhadas de Tamanho, Esforço, Prazo (Backlog do	Plano do projeto	Equipe de Desenvolvimento

(Continua)

		<i>Sprint</i>)		
Planejamento	Determinado conforme a reunião de <i>Backlog</i>	Criação do quadro <i>Kanban</i> e divisão das histórias (XP/SCRUM);	Plano do projeto	Equipe de Desenvolvimento
Planejamento	Determinado conforme planilha de esforço	Estabelecimento dos compromissos internos e externos do projeto	Plano do projeto	Coordenador Desenvolvimento
Planejamento	Determinado conforme planilha de esforço	Definição da estratégia de Monitoramento e Controle	Plano do projeto	Coordenador Desenvolvimento
Planejamento	Determinado conforme a reunião de <i>Backlog</i>	Elaboração do Plano de Comunicação, incluindo o planejamento das Reuniões de acompanhamento de projeto (<i>Daily Scrum</i>);	Plano do projeto	Equipe de Desenvolvimento
Planejamento	Determinado conforme planilha de esforço	Documentação Técnica	Documento Técnico	Desenvolvedores
Planejamento	Determinado conforme planilha de esforço	Criação de atividades 'macro' no quadro de tarefas		Coordenadores
Planejamento	Determinado conforme planilha de esforço	Criação atividades no <i>Redmine</i>		Coordenadores
Planejamento	Determinado conforme planilha de esforço	Atualização atividades / horas gastas no <i>Redmine</i>		Desenvolvedores QA Coordenadores
Planejamento	1 dia	Reunião de <i>kick off</i>	Ata de reunião	Desenvolvedores QA Coordenadores
Desenvolvimento	Determinado conforme planilha de esforço	Codificação dos programas / Desenvolvimento do Produto	Conjunto de programas desenvolvidos e documentados	Desenvolvedores
Desenvolvimento	Determinado conforme planilha de esforço	Elaboração das instruções de instalação e produção/operação	Manual de Instalação	Desenvolvedores
Desenvolvimento	Determinado conforme reunião de <i>Backlog</i>	Criação do código e algoritmo em pares (Dois programadores);		Desenvolvedores

(Continua)

Desenvolvimento	Determinado conforme reunião de <i>Backlog</i>	Daily Scrum (Diário);		Desenvolvedores
Desenvolvimento	Determinado conforme reunião de <i>Backlog</i>	Entrega das histórias em cada (<i>Sprint</i>);		Desenvolvedores
Desenvolvimento	Determinado conforme planilha de esforço	Execução de testes de unidade	Conjunto de programas desenvolvidos e documentados	Desenvolvedores
Desenvolvimento	Determinado conforme planilha de esforço	Revisão de prazo e esforço		Coordenador desenvolvimento
Desenvolvimento	Determinado conforme planilha de esforço	Revisão do plano de projeto		Coordenador desenvolvimento
Desenvolvimento	Determinado conforme planilha de esforço	Execução do monitoramento e controle do Projeto		Coordenador desenvolvimento
Desenvolvimento	Determinado conforme planilha de esforço	Realização de Reuniões de Acompanhamento de Projeto		Coordenador desenvolvimento
Desenvolvimento	Determinado conforme planilha de esforço	Criação casos de teste (<i>Testlink</i>)	Casos de teste no <i>Testlink</i>	QA
Desenvolvimento	Determinado conforme planilha de esforço	Atualização atividades / horas gastas no <i>Redmine</i>		Desenvolvedores QA Coordenadores
Desenvolvimento	Determinado conforme planilha de esforço	Acompanhamento do cliente		Coordenador desenvolvimento
Desenvolvimento	Determinado conforme planilha de esforço	Criação de <i>Release Note</i> para ambiente de QA	Release Note	Coordenador desenvolvimento
Testes	Determinado conforme planilha de esforço	Preparação de recursos para testes (programas, procedimentos, base de informações, instruções de instalação)		QA
Testes	Determinado conforme planilha de esforço	Preparação do ambiente de testes		QA
Testes	Determinado conforme planilha de esforço	Execução dos testes e atualização <i>bugzilla</i>		QA
Testes	Determinado conforme planilha de esforço	Verificação da integração com outros sistemas		QA

(Continua)

Testes	Determinado conforme planilha de esforço	Correção de <i>BUGS</i> / Verificações e atualização <i>bugzilla</i>	Conjunto de programas testados e documentados	Desenvolvedores QA
Testes	Determinado conforme reunião de <i>Backlog</i>	Daily Scrum (Diário)		Equipe
Testes	Determinado conforme reunião de <i>Backlog</i>	Atualização do quadro <i>Kanban</i> das histórias (XP/SCRUM)		Equipe
Testes	Determinado conforme planilha de esforço	Revisão de prazo e esforço		Coordenadores
Testes	Determinado conforme planilha de esforço	Revisão do Plano de projeto		Coordenador desenvolvimento
Testes	Determinado conforme planilha de esforço	Execução do monitoramento e controle do Projeto		Coordenador desenvolvimento
Testes	Determinado conforme planilha de esforço	Realização de Reuniões de Acompanhamento de Projeto		Coordenadores
Testes	Determinado conforme planilha de esforço	Criação de GMUD para ambiente de homologação	GMUD	Coordenador desenvolvimento
Testes	Determinado conforme planilha de esforço	Criação de <i>Release Note</i> para ambiente de homologação	<i>Release Note</i>	Coordenador QA
Testes	Determinado conforme planilha de esforço	Atualização atividades / horas gastas no <i>Redmine</i>		Desenvolvedores QA Coordenadores
Testes	Determinado conforme planilha de esforço	Criação de manuais (técnico e usuário)	Manuais (técnico e usuário)	Desenvolvedores QA
Homologação	Determinado conforme planilha de esforço	Preparação do ambiente de aceite		Desenvolvedores INFRA
Homologação	Determinado conforme planilha de esforço	Preparação dos recursos para aceite	Plano de Testes para homologação	QA
Homologação	Determinado conforme planilha de esforço	Realização de treinamento		QA
Homologação	Determinado conforme planilha de esforço	Acompanhamento dos testes do cliente pela equipe de QA		Coordenador QA
Homologação	Determinado conforme planilha de esforço	Solicitações de mudança para correção de problemas		Cliente QA

(Continua)

		detectados		
Homologação	Determinado conforme planilha de esforço	Execução de correções de problemas detectados		Desenvolvedores
Homologação	Determinado conforme planilha de esforço	Homologação do Sistema	Documento de aprovação	Cliente
Homologação	Determinado conforme planilha de esforço	Testes do <i>Check-List</i>		Cliente
Homologação	Determinado conforme planilha de esforço	Revisão do plano de projeto	Plano de Homologação	Coordenador desenvolvimento
Homologação	Determinado conforme planilha de esforço	Realização de Reuniões de alinhamento com o cliente	Ata de reunião	Coordenadores
Homologação	Determinado conforme planilha de esforço	Atualização atividades / horas gastas no <i>Redmine</i>		Desenvolvedores QA Coordenadores
Conclusão	Determinado conforme planilha de esforço	Criação de GMUD para ambiente de produção	GMUD	Coordenador desenvolvimento
Conclusão	Determinado conforme planilha de esforço	Criação de Release Note para ambiente de produção	<i>Release Note</i>	Coordenador QA
Conclusão	Determinado conforme planilha de esforço	Atualização atividades / horas gastas no <i>Redmine</i>		Desenvolvedores QA Coordenadores
Conclusão	Determinado conforme planilha de esforço	Apresentação do desenvolvimento realizado a equipe		Desenvolvedores

3.3. IMPLEMENTAÇÃO DA METODOLOGIA

Conforme explicado na introdução à página 48, era evidente que a empresa precisava de uma metodologia eficaz em seus projetos, com a perda de dinheiro e investimento, a empresa corria o risco de quebrar.

Para evitar tal situação, foram tomadas medidas que iriam não só melhorar a eficácia e qualidade de seus serviços como trazer de volta a ascensão da empresa. Sendo assim, foi realizada uma pesquisa interna na corporação, para que fossem identificadas as falhas existentes no processo de gestão.

Inicialmente, foi verificado que não havia dados para a pesquisa, deixando certo, que não haveria a possibilidade de se realizar métricas sem bases de dados existente. Com isso, após inúmeras reuniões, a solução encontrada foi criar uma equipe de pesquisa, que teria como principal atribuição desenvolver uma minuciosa pesquisa com o fito de identificar os problemas que obstavam o crescimento da empresa, estudar as principais metodologias disponíveis no mercado, fazer uma pesquisa sobre os costumes e cultura da empresa e as principais ferramentas de *softwares* para o caso em apreço.

3.4. IDENTIFICAÇÃO DOS PROBLEMAS

Para identificar os problemas que a empresa apresentava, a equipe de estudo destinada para tanto, optou por realizar um monitoramento em um projeto interno que já estava em execução, com prazo de entrega de aproximadamente de 8 meses.

Este monitoramento, dava-se através do acompanhamento do desenvolvimento do projeto, sem interferir no andamento do mesmo, pois o objetivo principal deste supervisionamento era justamente auferir falhas no decorrer do projeto.

Este projeto destinava-se a realizar a criação de *software*, que possuía a função principal de gerenciar o cadastro de usuários, bem como, a venda do produto principal do cliente.

A equipe iniciou o monitoramento logo no início deste projeto, sendo então possível identificar que o departamento responsável pela criação do *software* não seguia nenhum procedimento ou metodologia específica.

Verificou-se que cada integrante da equipe de desenvolvimento trabalhava de forma distinta e autônoma, utilizando apenas conhecimentos e experiências já adquiridos para o término da sua tarefa. Sendo assim, inicialmente, foi possível apurar que o departamento, como um todo, deveria utilizar uma metodologia específica e única para o desenvolvimento do *software*.

Outro erro evidenciado foi que a equipe, ao receber o projeto, já o recebia com uma data de entrega, sem ser realizada uma prévia consulta ou análise dos departamentos envolvidos no projeto para verificar se havia tempo hábil para cumprir o cronograma pré-estipulado.

Identificou-se também que a equipe de produtos – responsável pelo fornecimento das informações necessárias ao departamento de desenvolvimento – não as transmitia com especificidade e clareza, o que resultava no atraso do cumprimento do cronograma do projeto.

Ademais, era corriqueiro que houvesse a solicitação de mudança de algum requisito já informado ao departamento de desenvolvimento, o que resultava em uma revisão, e posterior readaptação, aos requisitos já implementados, sem que houvesse a alteração do cronograma inicial.

Em decorrência da mudança do cronograma inicial, sem prévio planejamento, foi possível apurar considerável aumento no orçamento da empresa dispendido com horas extraordinárias pagas aos colaboradores que, com o intuito de cumprir os prazos pré-estipulados, ultrapassavam a jornada de trabalho diária, permanecendo na empresa por até 16 horas por dia. Consequentemente, houve uma grande insatisfação por parte dos colaboradores, pelo fato de trabalharem excessivamente nos dias que antecediavam a entrega do projeto, juntamente, com a cristalina falta de organização e planejamento da empresa, gerando a perda de mão de obra qualificada.

Outro fator apresentado, era a ausência das principais documentações:

- Atas de reuniões;
- Documento de casos de teste;
- Documento de caso de uso;
- Documento de diagrama de sequencia;
- Documento de requisitos;
- Documento de visão;
- Documento de *wireframe*;
- Planos de teste, dentre outros mais.

Com a ausência destas documentações básicas, tornava-se ainda mais difícil a continuidade da execução do projeto quando havia a ausência de algum colaborador – por dispensa motivada ou imotivada –, colocando em risco a finalização do projeto, uma vez, que não se tinha a rastreabilidade das informações obtidas no início do projeto.

Sendo assim, quando havia a substituição deste, o novo colaborador tinha a missão de localizar os requisitos através de uma análise geral do projeto, de reuniões internas com a equipe de produtos, e posterior análise do código fonte.

Com a união de todas as falhas acima apresentadas, houve a perda de qualidade no *software* desenvolvido, gerando a manifesta insatisfação do cliente, os quais obtiveram um produto com condições inferiores daquele requerido anteriormente.

Mesmo após o término do monitoramento do projeto, ainda era possível ver os reflexos causados pela má gestão dos projetos, principalmente na área financeira da empresa que, em meio a demissões e contratações, horas extraordinárias e principalmente, orçamentos que ultrapassavam os valores provisionados iniciais, dispendiam valores excessivos e não provisionados.

Por fim, foi possível observar ainda, que as equipes não obtinham autonomia para a tomada de decisões relevantes e fundamentais pertinentes a sua área. E, que um dos fatores que atrasam o prosseguimento do projeto é, justamente a demora para a validação do que foi demandado para o responsável pela autorização – no caso em conteúdo, a própria diretoria da empresa. Ou seja, os departamentos ficam vulneráveis aguardando o retorno da aprovação.

3.5. CONSIDERAÇÕES FINAIS

Com o presente estudo de caso, foi possível analisar minuciosamente os percalços sofridos por uma empresa multinacional, que através de reiterados erros e falhas, estava apresentando prejuízos consideráveis, tanto na parte financeira, quanto na parte de qualidade de seus produtos.

Dessa forma, após diversas reuniões e pesquisas realizadas, foram apontadas e evidenciadas as falhas encontradas durante todos os trâmites internos da companhia, através de um acompanhamento severo realizado pela equipe de pesquisas com o monitoramento dos projetos já em curso.

Após a equipe de pesquisa realizar os monitoramentos, foram apuradas as falhas e estudos dos melhores mecanismos para que fosse possível sanar os problemas apresentados. Para tanto, utilizou-se uma metodologia nova, embasada em três outras já existentes no mercado, a qual foi devidamente adaptada e regulamentada aos costumes e à cultura interna da empresa, visando à maior efetividade da metodologia.

Inicialmente, os novos procedimentos foram implementados em apenas um departamento, e após a real e funcional adaptação do setor, bem como de seus colaboradores, a nova metodologia foi sendo expandida e difundida aos poucos para todos os outros setores da empresa.

Com essas novas mudanças, em pouco tempo, a empresa obteve consideráveis evoluções internas nos departamentos que aderiram a nova metodologia e, conseqüentemente, a empresa auferiu lucros maiores do que os provisionados para aquele período.

A utilização desta nova metodologia foi tão eficaz que somente após um ano de implantação a empresa alcançou renomada colocação no mercado em que atua, evidenciando que é possível contornar uma má gestão de processos, utilizando metodologias moldáveis à cultura da empresa.

4. MÉTRICAS

Para que seja possível demonstrar a viabilidade da implantação da nova metodologia, as problemáticas identificadas na empresa em apreço, serão apresentados dados verídicos adquiridos através do estudo de caso.

Para facilitar o entendimento, será feita uma subdivisão das métricas elaboradas, sendo: pessoas, projetos e qualidade.

4.1. PESSOAS

Neste tópico será demonstrado que, após a implantação da nova metodologia na empresa, as problemáticas apresentadas, no que tange a demissões e perda de colaboradores, foram devidamente solucionadas.

Utilizando parâmetros básicos e com a adequação das horas trabalhadas, o uso da nova metodologia também beneficiou os colaboradores no que se refere às horas trabalhadas, pois com maior organização e uma boa gestão de processos, houve a diminuição das horas extraordinárias. Em função dos novos procedimentos, os cronogramas começaram a ser fielmente seguidos, não havendo a necessidade de desgaste dos colaboradores com expedientes estendidos.

As horas extras, sem sombra de dúvida, eram um grande fator prejudicial à companhia, inicialmente pelos gastos despendidos e não provisionados. Além

destes gastos, eram acrescidas despesas “auxiliares” às horas, que além de acontecerem em período noturno – para o qual o pagamento é ainda mais dispendioso – vinham acompanhadas com pagamento de táxi para os colaboradores retornarem para casa, na maioria das vezes, nas altas horas da madrugada, e também de refeições realizadas durante o período extraordinário.

Todos estes fatores juntos, se tornam demasiadamente onerosos em larga escala para a empresa.

Fora os elevados custos para a corporação, ocorreu também a sobrecarga dos colaboradores, que em dias que antecedia a entrega do projeto, trabalhavam excessiva e exaustivamente. Sendo assim, era evidente que em um período de curto de prazo, haveria a desmotivação e insatisfação dos membros da equipe desenvolvidora do projeto, resultando em atrasos, falta de interesse, e que, em muitas vezes, ocasionou o desligamento do colaborador.

Pode-se perceber que a empresa não possuía uma metodologia eficaz e acabava tendo desperdícios e perdas tanto na qualidade do produto que oferecia, quanto com a saída colaboradores capacitados, porém, insatisfeitos.

Outro aspecto importante, que leva a compreender o progresso dessa metodologia, dá-se na não-centralização das informações.

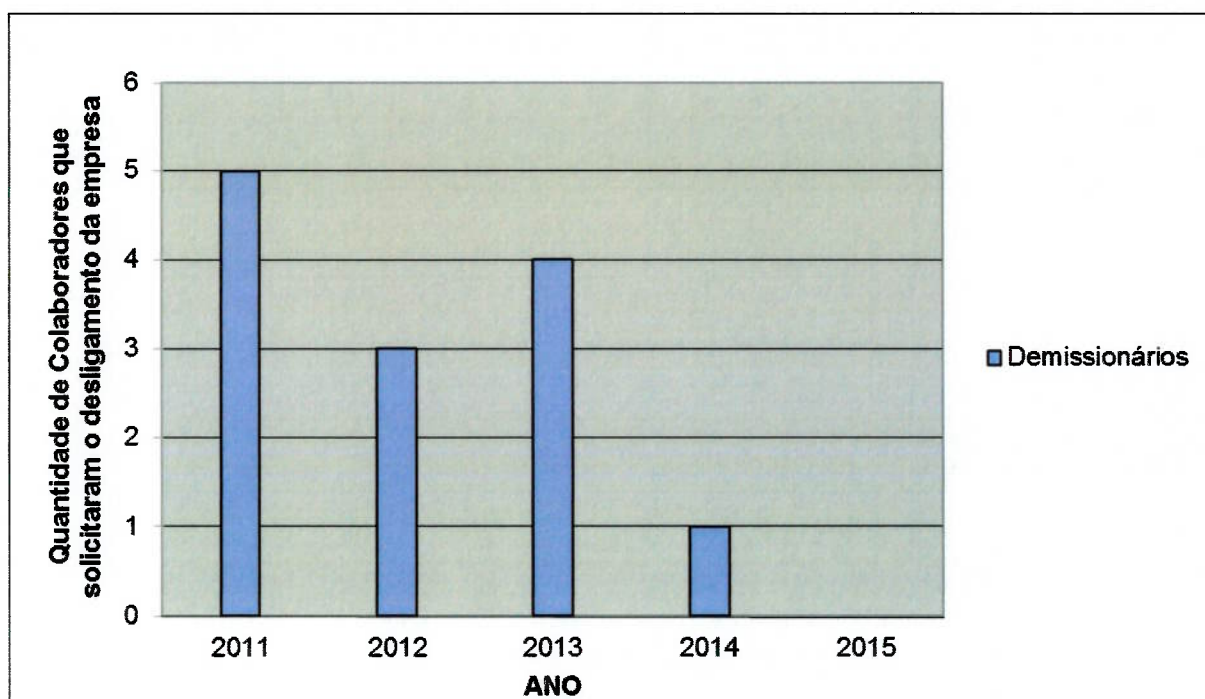
Antes da utilização da nova metodologia as informações eram centralizadas e pouco divulgadas e as equipes ficavam à mercê de uma aprovação.

Com os novos procedimentos, cada departamento possui sua própria autonomia, pois é feita uma apresentação do que será desenvolvido. Esta tática motivou os colaboradores, que com mais liberdade e independência, se sentiram mais prestativos e com voz ativa dentro de sua equipe, gerando uma satisfação interna.

Esta autonomia das decisões, incentivou de forma contagiante os departamentos da empresa e tornou-se um importante instrumento, inclusive para os novos colaboradores, pois no passado, muitos estavam insatisfeitos.

Atualmente, com o bom andamento e resultados positivos dos projetos, os colaboradores fazem importante marketing da empresa para amigos e colegas, pois a mesma tornou-se referência na área em que atua, agora, com excelência.

Segue, abaixo, gráfico comparativo do quadro de colaboradores antes, durante e após a implantação da nova metodologia, demonstrando a satisfação dos integrantes da empresa.

Gráfico 1 - Demissionários

É evidente que uma companhia perde dinheiro quando existe uma elevada rotatividade de colaboradores e com a utilização da nova metodologia implantada, foi possível verificar que os colaboradores estavam insatisfeitos com a maneira que a empresa realizava seus projetos.

Importante salientar que, para que esta implantação fosse eficiente e atingisse os objetivos propostos, foi necessário que houvesse inúmeras palestras, reuniões com os gestores, treinamentos e provas práticas para alinhar os novos processos utilizados, aplicados paulatinamente e com cautela, pois não seria possível – ou levaria mais tempo – para reverter o quadro da empresa no que tange à permanência dos colaboradores, como se pode notar através do gráfico 1.

4.2. PROJETOS

No presente item, será demonstrado a quantidade de projetos que foram realizados antes, durante e após a implantação da nova metodologia na empresa.

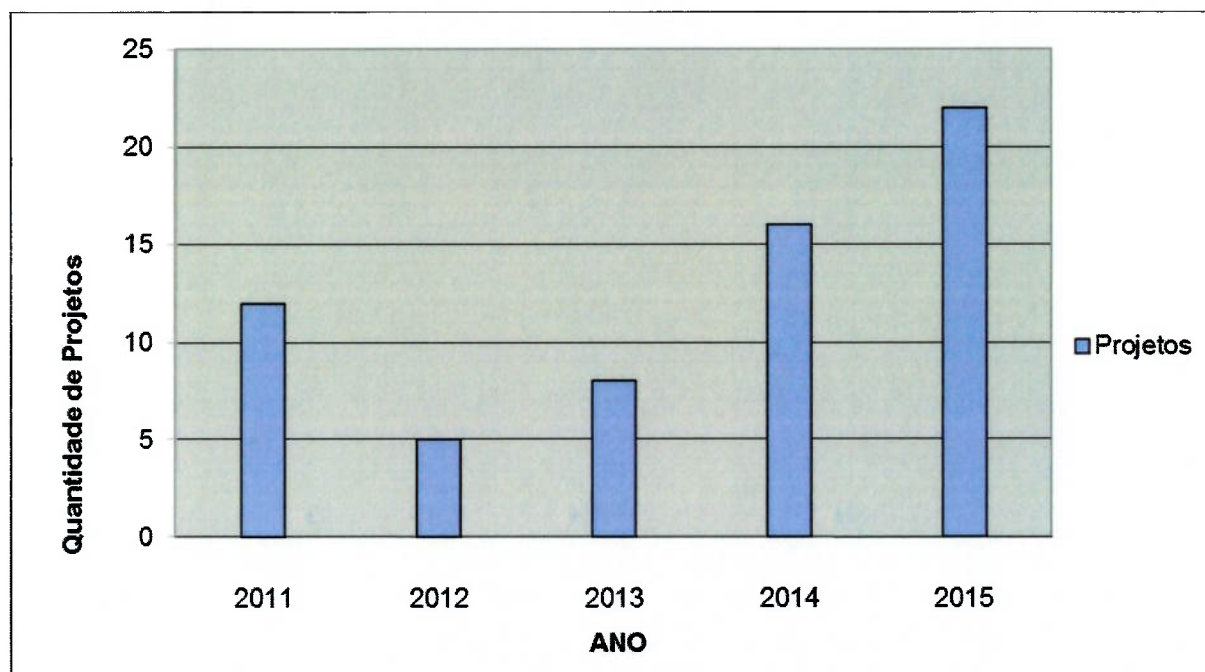
É possível identificar a evolução dos projetos, tanto no quesito qualidade – o qual abarca, praticamente, todas as fases do projeto –, como no quesito de cumprimento tempestivo do cronograma.

É notório que a cultura da empresa mudou totalmente em prol de melhorias e através de novos procedimentos implantados, pois com um processo pré-definido os membros da equipe já possuem conhecimento das tarefas que devem realizar e os procedimentos iniciais que devem ser tomados durante todo o desenvolvimento do projeto.

Conforme demonstra o gráfico 2, é possível identificar que no ano 2011 – logo após o nascimento da empresa – houve vários projetos a serem desenvolvidos. Contudo, com a inexistência de uma metodologia eficaz houve falhas durante o desenvolvimento dos projetos, conforme exaustivamente mencionados em tópicos anteriores, consequentemente, no ano seguinte, a empresa em apreço obteve uma expressiva queda de solicitações de projetos.

Somente após a implantação da nova metodologia – em meados de 2013 – foi possível começar a reverter um quadro de declínio para ascensão, pois em 2015, foram desenvolvidos mais que o dobro de projetos realizados em 2013.

Gráfico 2. - Desenvolvimento dos projetos antes, durante e após a implantação da nova metodologia



5. QUALIDADE

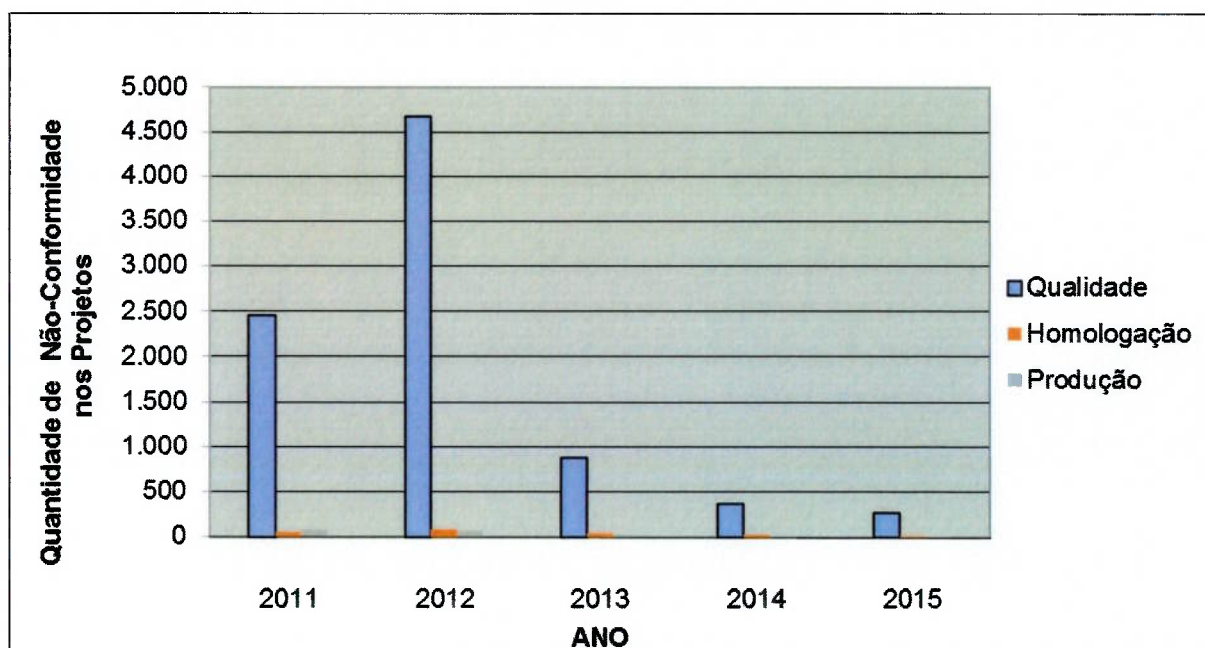
No gráfico 3, será devidamente demonstrada a evolução da qualidade dos projetos desenvolvidos pela empresa, antes, durante e após a implantação da nova metodologia.

Sendo assim, pode-se verificar que em 2011 e 2012 houve grande quantidade de não-conformidades encontradas pela equipe de qualidade durante a realização dos testes.

Neste mesmo período, com a ausência de uma metodologia eficaz, a equipe de qualidade, ao realizar os testes, não identificava algumas falhas que só eram verificadas ou pelo cliente, no ambiente de homologação, ou através do próprio cliente final, ao utilizar o *software* no ambiente de produção. Esse efeito ocasionava perda financeira para empresa e perda de qualidade para o cliente.

Ademais, após a implantação da nova metodologia, em meados de 2013, já é possível observar modificações positivas durante a realização de testes, ou seja, com os novos processos utilizados pela empresa, a equipe de desenvolvimento passou a entregar o sistema, com a maioria dos requisitos solicitados pelo cliente, com qualidade superior as àquelas apresentadas antes da utilização da nova metodologia.

Além do mais, em 2015 é notável a ausência de não-conformidades, pois a equipe de qualidade entregava os sistemas com os requisitos de acordo com aqueles definidos no início do projeto, e assim, o departamento de qualidade ao receber o sistema podia prosseguir com o desenvolvimento de suas funções com mais segurança e confiabilidade nos procedimentos já realizados, diminuindo a margem de não-conformidades apresentadas nos anos anteriores à utilização da nova metodologia.

Gráfico 3 - Não conformidades por ano

6. CONCLUSÃO

A presente monografia teve como objetivo demonstrar que é possível reverter um quadro de declínio financeiro de uma empresa que não utilizava métodos eficazes para gerir os seus processos internos.

Em decorrência dessa má gestão dos processos, houve perdas significativas não só da lucratividade da empresa, mas também de capital intelectual – mão de obra qualificada –, investidores, gastos demasiadamente onerosos e sem prévio provisionamento, além da perda de credibilidade no mercado em que atua.

Sendo assim, para contornar este panorama crítico, a corporação optou, após detalhados estudos internos, pela utilização de uma metodologia competente, e com a colaboração de todos os membros, notou que seria possível modificar o desempenho da empresa.

Para tanto, foi desenvolvido um departamento de pesquisa que ficou responsável de identificar os maiores problemas apresentados na corporação, bem como estudar as melhores soluções cabíveis para as falhas encontradas. E, após o consenso da equipe, concluíram que era possível utilizar três metodologias já existentes – SCRUM, Extreme Programming (XP) e PMBOK – e moldá-las, juntas, à cultura e

aos costumes da empresa, visando sanar as dificuldades identificadas na pesquisa anteriormente realizada.

Estas três metodologias – que embasaram o presente estudo de caso – unidas, viabilizam uma gestão eficaz com os novos procedimentos implementados, com o intuito de potencializar a qualidade dos produtos finais, organizar e padronizar o desenvolvimento dos projetos até a efetiva entrega deste ao cliente.

Com as métricas elaboradas, foi possível notar que, após a implementação desta nova metodologia, ajustada aos costumes da empresa, houve diversas melhorias nos departamentos da corporação. No que tange aos colaboradores, estes foram diretamente beneficiados com a implantação desta metodologia, pois passaram a ter uma coordenação eficaz, com cronogramas estruturados e compatíveis com o real desenvolvimento do projeto. Isso ocasionou a reversão de um quadro de insatisfações e demissões.

Em relação à solicitação de criação de novos projetos, houve notória ascensão após a implantação, e atualmente, os departamentos cumprem os prazos estipulados, atendem rigorosamente os requisitos estabelecidos pelos clientes e garantem a qualidade dos produtos desenvolvidos pela empresa.

Além do mais, no quesito Qualidade, a mudança de procedimento foi tão benéfica, que os desenvolvedores, ao realizarem os testes unitários, verificaram a queda das falhas e não-conformidades existentes nos *softwares* desenvolvidos, deixando certo que a nova metodologia atingiu não só os objetivos traçados inicialmente, mas também fomentou os mecanismos internos relacionados às equipes e departamentos da empresa.

Sendo assim, com o auxílio das métricas apresentadas em capítulos anteriores, são evidentes as melhorias e os avanços que foram alcançados através da nova metodologia, levando a empresa em apreço não só a contornar seus percalços, como também à ascensão no mercado de certificação digital.

Ademais, neste trabalho, além de comprovar o progresso da Certified Inovation S.A., através da utilização de uma nova metodologia, demonstra que, para uma metodologia atinja seus reais objetivos, é necessário ir além da simples implementação desta. É preciso que haja uma minuciosa e específica pesquisa interna, identificando falhas e problemas, para que, então, esta nova metodologia, seja devidamente moldada e adaptada, adequando-se aos costumes e à cultura da empresa, atendendo às funcionalidades que lhes forem pertinentes.

7. REFERÊNCIAS BIBLIOGRÁFICAS

AGILE MANIFESTO. Disponível em <<http://agilemanifesto.org>>. Acesso em 02 fev. 2016.

BECK, K. **Extreme Programming explained: Embrace Change.** Edição US. Boston: Addison-Wesley, 2000.

BROD, Cesar. **Scrum: Guia prático para projetos ágeis.** São Paulo Novatec Editora, 2013.

CANEDO, E. **Introdução ao Gerenciamento de Projetos.** Brasília: [s.n.], 2007. Apostila utilizada no MBA da UNIEURO.

FERNANDES, J. H. C.; RALHA, C. G. **Uma Introdução ao Gerenciamento de Serviços em Organizações de TI**, baseada no Modelo Information Technology Infrastructure Library – ITIL. 2005

FOWLER et al. **Refatoração: Aperfeiçoando o Projeto de Código Existente.** 1ª edição. [S.l]: Bookman, 2004.

JEFFRIES, R. What is Extreme Programming. **XProgramming.com - An Agile Software Development Resource**, 1999. Disponível em: <<http://xprogramming.com>>. Acesso em: jul. 2016.

KERZNER, HAROLD [recurso eletrônico], **Gestão de projetos: as melhores práticas; tradução Lene Belon Ribeiro.** - 2. ed. - Dados eletrônicos, Porto Alegre, Bookman, 2007.

KERZNER, H. (2006). **Project management: A systems approach to planning, scheduling, and controlling.** Hoboken, NJ: John Wiley & Sons, Inc.

KNIBERG, Henrik. **Scrum e XP direto das Trincheiras.** C4Media, 2007. Disponível em: <<http://www.infoq.com/br/minibooks/scrum-xp-from-the-trenches>>. Acesso em: 02 fev. 2016.

MARÇAL, P.; PEREIRA, P.; TORREÃO, A. S., **Entendendo Scrum para Gerenciar Projetos de Forma Ágil**, Rio de Janeiro, 2007.

MARTINS, JOSÉ C. C., **Gerenciando projetos de desenvolvimento de software com PMI, RUP e UML.** 5. ed. Rio de Janeiro, Brasport, 2010.

MARTINS, JOSÉ C. C., **Técnicas para gerenciamento de projetos de software.** Rio de Janeiro: Brasport, 2007.

MARCHESI, M., SUCCI, G., WELLS, D. e WILLIAMS, L. **Extreme Programming Perspectives.** Michigan: Addison Wesley Professional, 2002.

MAXIMIANO, A. C. A. Administração de projetos: **Como transformar ideias em resultados.** São Paulo: Atlas, 2 ed. 2002.

NOGUEIRA, MARCELO, **Um framework para a gestão de Riscos em Projetos de Software**, Dissertação de Mestrado, UNIP, São Paulo, 2004.

POPPENDIECK, Mary; POPPENDIECK, Tom. **Lean software development: an agile toolkit**. 1. ed. Upper Saddle River, NJ: Addison-Wesley, 2003

PRESSMAN, ROGER S. **Engenharia de Software: Uma Abordagem Profissional** 7. ed. CGRAW HILL - ARTMED, 2009.

PROJECT MANAGEMENT INSTITUTE. **Um guia do conhecimento em gerenciamento de projetos (Guia PMBOK®)** 5 ed. - São Paulo: Saraiva, 2014

SCHWABER, Ken; SUTHERLAND, Jeff. Scrum Guide. Scrum.Org, Fevereiro 2010.
Disponível em: < <http://www.scrum.org/scrumguides/>>. Acesso em: 02 fev. 2016.

TELES, M. Vinícius. **Extreme Programming: Aprenda como encantar seus usuários desenvolvendo software com agilidade e alta qualidade**. 2º Edição, São Paulo, Novatec Editora, 2014.

VARGAS, Ricardo Vianna. **Gerenciamento de Projetos - Estabelecendo Diferenciais Competitivos**. 7 ed. Rio de Janeiro, 2009. 236p.

VARGAS, R. V. **Manual Prático do plano do projeto: utilizando o PMBOK guide**. 3ª ed. Rev., Brasport, Rio de Janeiro, 2007.

VERZUH, E. **The fast foward in MBA in project management**. John Wiley & Sons, Inc., 1999