GUILHERME VERNINI PONTES

APLICAÇÃO DO CICLO PDCA NA OTIMIZAÇÃO DO PROCESSO DE ELABORAÇÃO DE SOFTWARES NA ÁREA DE TECNOLOGIA DE UM BANCO DE VAREJO

São Paulo

2012

GUILHERME VERNINI PONTES

APLICAÇÃO DO CICLO PDCA NA OTIMIZAÇÃO DO PROCESSO DE ELABORAÇÃO DE SOFTWARES NA ÁREA DE TECNOLOGIA DE UM BANCO DE VAREJO

Monografia apresentada à Escola Politécnica da Universidade de São Paulo para obtenção do certificado de Especialista em Gestão e Engenharia da Qualidade – MBA/USP

Orientador:

Prof. Dr Adherbal Caminada Netto

São Paulo

2012

DEDICATÓRIA

Dedico este trabalho aos meus familiares e minha namorada, em especial minha mãe pelo suporte incondicional.

AGRADECIMENTOS

Agradeço minha família pelo apoio e compreensão, em especial minha mãe.

Ao orientador do curso Prof. Dr. Adherbal Caminada Netto pela orientação para a realização desta monografia, e aos professores e funcionários que contribuem para a realização do curso de Gestão e Engenharia da Qualidade da Universidade de São Paulo.

Agradeço aos amigos e colegas, pelo incentivo e apoio constantes.

O que sabemos é uma gota, o
que ignoramos é um oceano

(Isaac Newton)

RESUMO

O escopo do trabalho será a aplicação do ciclo PDCA com auxílio de ferramentas de qualidade (Brainstorming, Ishikawa, Matriz de priorização, etc) para mapear qual seria a causa-raiz das não-conformidades(Erro lógico de programação, prazo de entrega não cumprido, não cumprimento de requisito de escopo, não atendimento a diretriz corporativa da arquitetura de dados, identificação de inconsistências nos cálculos, etc) identificadas nos softwares entregues para área cliente interna, as causas de não conformidades são relatadas pelos clientes internos nas pesquisas de satisfação juntamente com a análise de indicadores de qualidade. A identificação da causa raiz permite que decisões de cunho gerencial possam ser tomadas, ou seja, é possível que o ponto crítico , aquele que possui maior interferência na qualidade final da entrega, possa ser tratado para melhorar os resultados.

Palavras-chave: Ferramentas da Qualidade. Elaboração de Softwares. Melhoria Contínua.

ABSTRACT

The scope of this work will be the application of the PDCA cycle with the help of quality tools(Brainstorming, Ishikawa, Histogram, Prioritization Matrix) in order to map the root cause of non-conformities(Programming logical error, missed deadline, scope requisite not met, no compliance with the corporate data architecture guideline, calculation inconsistencies) found in the software delivered to the internal customer, the non-conformities causes are told by the internal customers in the satisfaction surveys and with the analysis of quality indicators.The root cause identification allows managerial decisions to be made, in other words, it is possible that the critical point, which has the greater interference in the final deliver, can be dealt with in order to improve the results.

Keywords: Quality tools. Software Development. Continuous Improvement.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

PDCA	Plan,Do,Check,Act
ISO	International Organization for Standardation
TI	Tecnologia da Informação
PMBOK	Project Management Body Of Knowledge
SLA	Service Level Agreement

SUMÁRIO

1. INTRODUÇÃO

1.1 HISTÓRIA

Esta monografia tem como foco a Instituição Financeira X, que possui sua matriz localizada na cidade de São Paulo, atua no segmento financeiro, a empresa possui mais de 5 mil funcionários o que nos permite categorizá-la como empresa de grande porte.

A Instituição Financeira X possui uma área de TI (Tecnologia da Informação) responsável por elaborar softwares, implementar atualizações e dar suporte para Área de Negócios (Cliente Interno), o processo que será fruto do estudo desta monografia será a elaboração de softwares que é realizada pela Área de TI (Tecnologia da Informação).

O setor bancário brasileiro é extremamente competitivo, o processo produtivo do setor bancário possui suas particularidades, dado que se enquadra numa categoria de prestadores de serviços, as matérias-primas necessárias para elaboração de seu produto final(serviço) são os recursos humanos e ferramentas de trabalho.

Segundo CASSIOLATO (1992) e FRISCHTAK (1992), as características do setor, como dispersão geográfica das agências, concentração de conglomerados financeiros e a parceria estabelecida com produtores de informática, constituíram-se nas causas iniciais do processo de automação bancária. Este processo se iniciou na década de 1970 por iniciativa do setor privado. No primeiro estágio , a ênfase da automação foi administrar e controlar as operações. O segundo, a partir da década de 80, foi centrado em políticas de diferenciação de produtos e serviços, ou melhor, em inovações financeiras associadas à informática e centrada no cliente. Esse processo constitui-se claramente em aumentos de produtividade 1979/88 , tornando o setor bancário um dos mais rentáveis da atividade econômica.

Conforme DIEESE (1997), o ambiente com alto grau de incerteza, competitivo e de estabilidade econômica, desde a implantação do Real, foi colocando para o setor bancário a necessidade de fazer mudanças bruscas em sua estrutura operacional tendo como eixo principal o trinômio, demissão em massa, automação e terceirização. Estas mudanças ocorreram em momentos diferenciados do ajuste estrutural implementado pelos bancos.

O processo de automação bancária foi um dos fatores que fez com que houvesse gradual substituição da mão de obra humana por equipamentos, sendo assim um dos fatores que desencadearam demissão em massa no setor bancário (Utilização do Internet Banking, criação caixas eletrônicos, etc), mas com uma análise mais aprofundada observamos que na verdade houve uma substituição de mão de obra (Caixa de banco, Auxiliar Administrativo) que realizavam transações bancárias simples(Saque, Depósito, Transferências, etc) por uma mão de obra especializada(Analistas de Tecnologia da Informação, Gerentes de Tecnologia da Informação, Programadores) que realizam atividades mais complexas de elaboração de softwares, criação e manutenção de funcionalidades no bankline, criação de softwares bancários para utilização em dispositivos móveis.

A problemática citada pelo autor de demissão em massa e terceirização, muito se deve, no contexto atual, ao fato das instituições financeiras buscarem de modo incessante a maior lucratividade possível e isto recai em diminuição de custos, se a produtividade do profissional está abaixo do esperado, este profissional é demitido, vale lembrar aqui que o **turnover** [1] das instituições financeiras é um dos mais altos que existem e normalmente esta substituição de mão de obra ocorre com diminuição da remuneração média, conforme tabela abaixo:

Remuneração Média (em R$)	Masculino	Feminino	Diferença em % Remuneração Média
Admitidos	2.842,18	2.121,72	-25,35%
Desligados	4.644,93	3.368,66	-27,48%

Tabela 1- Remuneração Média dos admitidos e desligados, Jan a Jun de 2011
Fonte: MTE Cadastro Geral dos Empregados e Desempregados (Lei 4923/65)

[1] **Turnover** = Rotatividade de pessoal, estabelece uma relação entre a quantidade de admissões e demissões para um dado intervalo de tempo.

A mão de obra terceirizada é utilizada para realização de atividades que exigem menor conhecimento técnico, situação para a qual as instituições financeiras tem evitado a utilização de um recurso do banco com conhecimento técnico bem aprofundado pois a hora de trabalho do recurso interno do banco é bem mais cara e este recurso é utilizado internamente para execução de uma tarefa mais complexa.Ainda levando em consideração a estratégia gerencial de diminuição de custos, foi verificado em 2012 que a media salarial dos funcionários demitidos era cerca de 30% superior a dos funcionários recém contratados.

A crescente importância do segmento de TI para as instituições financeiras pode ser observada através da constatação que a proporção de funcionários de TI em relação à quantidade total de funcionários de um banco tem aumentado significativamente, e os investimentos do setor bancário em TI são os mais elevados quando comparados a qualquer outro setor da economia.

1.2JUSTIFICATIVA

O segmento de TI das instituições financeiras tem sido ao longo dos últimos anos uma das principais prioridades operacionais e tem recebido grandes investimentos pois trata-se de um segmento extremamente competitivo e caso seja tratado com devida prioridade pode trazer vantagens competitivas.

Para PORTER (1995) , a transformação tecnológica não é, por si só, importante, mas é importante se afetar de uma forma significativa a vantagem competitiva de uma empresa.

Segundo MEIRELLES (2009) as empresas brasileiras investiram cerca de 6% do faturamento líquido em TI no ano de 2008. O setor de comércios investe cerca de 2,7% dos teus rendimentos em TI, o setor de indústrias 4% e o setor bancário lidera com 12%.

POSIÇAO	EMPRESA
1º	Petrobrás
2º	***Itaú Unibanco Holding***
3º	***Banco Bradesco***
4º	***Banco do Brasil***
5º	Vale
6º	Itaúsa
7º	Eletrobrás
8º	Companhia Siderúrgica
9º	Cemig
10º	Tele Norte Leste (OI)

Tabela 2- Relação das empresas de maior lucro em 2012
Fonte: FORBES(2012)

Ainda segundo a FORBES (2012), a tabela acima representa as 10 empresas que obtiveram maior lucro em 2012 no Brasil, em negrito estão destacadas as instituições financeiras.

Conforme as fontes de informações citadas foi constatado investimento massivo do faturamento líquido das empresas do setor bancário em TI e dentre as 10 empresas de maior faturamento liquido em 2011 foram observadas 3 empresas do setor bancário, o que por sua vez nos permite concluir que tal postura foi um dos fatores que propiciaram vantagem competitiva para empresas do segmento bancário, mais do que isto, se tornou um requisito para assegurar a competitividade de uma empresa do setor bancário frente as demais.

Conforme PÁDUA e MENDES (2009) um estudo demonstra a imaturidade das indústrias de software através dos seguintes indicadores:

- Mais de 30% dos projetos são cancelados antes de serem finalizados.
- Mais de 70% dos projetos falham nas entregas das funcionalidades esperadas.
- Os custos extrapolam em mais de 180% os valores originalmente previstos.
- Os prazos excedem em mais de 200% os cronogramas originais.

Importante ressaltar aqui que para projetos de outros segmentos: setor automotivo, setor aeronáutico, setor de construção civil, telefonia celular estes indicadores não são tão críticos, o percentual de cancelamento não é tão elevado, o percentual de projetos que falham não é tão elevado, até porque este parâmetro esta diretamente relacionado a satisfação do cliente, isto tudo reforça a análise dos autores no que se refere a imaturidade das industrias de software.

Conforme KIERAS (2009), professor de Ciências da Computação na Universidade de Michigan, os softwares não tem sido elaborados com maior qualidade devido a problemas não técnicos na etapa de desenvolvimento, dentre os quais destaca freqüente comunicação ineficiente entre desenvolvedores e área solicitante resultando em softwares elaborados sem corresponderem exatamente o que a área solicitante tinha em mente, o processo através do qual os softwares são desenvolvidos não ajudam só pioram este quadro, o autor ainda ressalta que se não houver um processo instaurado e organizado na empresa para elaboração de software a probabilidade de insucesso pode ultrapassar 50%.

Deste modo podemos observar que a indústria de software ainda está com muitas falhas, processos não eficientes, ou seja, passível de aplicação das ferramentas da qualidade e de obtenção de melhores resultados. Portanto a tentativa de melhoria de processos se torna justificável.

1.3 OBJETIVO E LIMITAÇÕES

O objetivo deste trabalho é aprimorar o processo de elaboração de softwares da **Instituição Financeira X** com o intuito de diminuir a quantidade de não-conformidades , para tal será feita a aplicação das ferramentas de qualidade (Brainstorming, Ishikawa, Matriz de priorização, etc) para mapear qual seria a causa-raiz das não-conformidades(Erro lógico de programação, Requisito de Negócio não atendido, Requisito de Técnico não atendido, Indisponibilidade Ambiente de Homologação , Erro parametrização Ambiente, etc) identificadas nos softwares entregues para Área de Negócios(Cliente interno).

As causas de não conformidades são relatadas pela Área de Negócios(Cliente interno) nas pesquisas de satisfação juntamente com a análise de indicadores de qualidade, para este trabalho haverá uma forma complementar de levantamento de informações que consistirá no envio e posterior análise de formulários pré-formatados.

Por fim após a identificação da causa raiz será aplicado os planos de ações cabíveis, ou seja, é possível que o ponto crítico, aquele que possui maior interferência na qualidade final da entrega, possa ser tratado para melhorar os resultados.

Dentre as principais limitações podemos ressaltar restrição de recursos financeiros, humanos e materiais.

A principal premissa é que o método seja aplicado levando em consideração suas particularidades e limitações, pois a incorreta aplicação do método levará a resultados distorcidos e conclusões precipitadas.

1.4 METODOLOGIA

Na elaboração desta monografia, foi utilizado como referência o acervo da Biblioteca de Engenharia Mecânica e Naval, assim como diversos sites da internet e livros da área de Qualidade.

Para identificar o problema de maior gravidade, aquele que possui maior interferência na qualidade final do produto(Elaboração de software) foi enviado um formulário para a Área de Negócios(Cliente Interno) para que fosse preenchido com as principais não-conformidades identificadas no processo de Homologação dos softwares elaborados.

Com o recebimento dos formulários preenchidos pela Área de Negócios(Cliente Interno) foi aplicada a ferramenta Matriz de priorização que permitiu elencar a não-conformidade mais crítica.

Com a identificação da não-conformidade mais crítica foi realizado um ***Brainstorming***[2] e ***Ishikawa***[3] com o intuito de definir quais seriam as prováveis causas da não- conformidade mais crítica.

Com a identificação das causas, foi feita uma análise levando em consideração restrições financeiras, humanas e de materiais para que pudesse ser traçado um plano de ação.

Após a execução do plano de ação foram enviados os formulários novamente com recorrência quinzenal para permitir o monitoramento e medição da quantidade de não- conformidades.

---

[2] **Brainstorming** = "Tempestade de idéias", é uma técnica de trabalho em grupo para geração de ideias.

[3] **Ishikawa** = Autor da ferramenta da Qualidade conhecida por Diagrama de Causa e Efeito ou simplesmente Ishikawa em homenagem ao autor.

2 REVISÃO BIBLIOGRÁFICA

2.1 FERRAMENTAS DA QUALIDADE

De acordo com MARSHALL e JUNIOR et AL (2006) na década de 1950 houve o surgimento das ferramentas de qualidade com base nos conceitos e práticas existentes naquela época, desde então estas ferramentas tem sido utilizadas nos sistemas de gestão da Qualidade, e são utilizadas para identificação de problemas, otimização e melhoria contínua dos processos.

As ferramentas de Qualidade estão por muitas vezes enraizadas em cálculos estatísticos, o que nos permite utilizá-las para medição, monitoramento e melhoria da qualidade de processos.

Nesta monografia serão utilizadas as ferramentas de Qualidade: Matriz de priorização, ***Brainstorming*** e Diagrama de Causa e Efeito.

2.1.1 Brainstorming

Conforme manual do SEBRAE (2005) tal ferramenta gerencial da Qualidade é de autoria de Alex Osborn que a divulgou em 1938. A tradução literal de ***Brainstorming*** é "Tempestade de Idéias", que no fundo nada mais é do que uma técnica de trabalho em grupo para geração de idéias.

Normalmente o ***Brainstorming*** é dividido em duas etapas a primeira Divergente, onde cada participante emite idéias de forma livre, sem que sejam feitas criticas, gozações ou qualquer tipo de julgamento, já a segunda etapa é a Convergente, é feita a análise onde o grupo elimina todas as idéias absurdas referentes ao tema, depois é feita a compactação que consiste em agrupar as idéias semelhantes e por fim a classificação onde o grupo atribui um grau de importância as idéias.

Segundo o autor Alex Osborn os princípios que norteiam o ***Brainstorming*** são:

- Atraso do julgamento
- Criatividade em quantidade e qualidade

O atraso de julgamento permite que boas idéias não sejam desprezadas precocemente e o segundo principio da criatividade em quantidade e qualidade parte do pressuposto estatístico de quanto maior for a quantidade de idéias, numa amostra não tendenciosa, maior será a possibilidade de encontrar uma idéia de qualidade.

Ainda conforme manual do SEBRAE (2005) existem algumas orientações que possibilitam aumentar a probabilidade de sucesso de um ***Brainstorming:***

- Quantidade de participantes entre 4 e 10
- Curta duração cerca de 1 hora
- Condições de ambiente adequadas: iluminação, temperatura, etc.
- Ter um facilitador (Se possível, profundo conhecedor do processo)
- Enfatizar as regras necessárias para realização de cada etapa
- Certificar que todos tenham entendido o tema

Por constatação pessoal do autor da monografia, em sessões de ***Brainstorming*** realizadas na área de software de gestão empresarial, TI segmento bancário e construção civil, se a quantidade de participantes for inferior a 4, muitas vezes a quantidade de idéias construtivas e por conseqüência a qualidade da sessão de ***Brainstorming*** pode não atingir o resultado esperado, já para os casos em que a sessão possuir muito mais do que 10 participantes pode ocorrer certa dificuldade de organização e também trazer dificuldades para que o resultado esperado seja atingido.

Para elaboração desta monografia foram realizadas diversas sessões de ***Brainstorming,*** para todas houve a preocupação de contar com a participação de pelo menos um profissional que possuísse profundo conhecimento do processo de elaboração de software para que o mesmo pudesse contribuir de forma efetiva na etapa de Convergência onde as idéias absurdas eram eliminadas.

2.1.2 Matriz de Priorização

Conforme HEGEDUS (2003) a matriz de priorização se aplica no seqüenciamento de itens conforme os critérios estabelecidos.

O diagrama de matrizes pode ser uma matriz do tipo L, matriz do tipo T, matriz do tipo X ou matriz do tipo C, a matriz de priorização é um caso especifico de diagrama de matrizes que relaciona apenas dois fatores, deste modo é representado por uma matriz do tipo L.

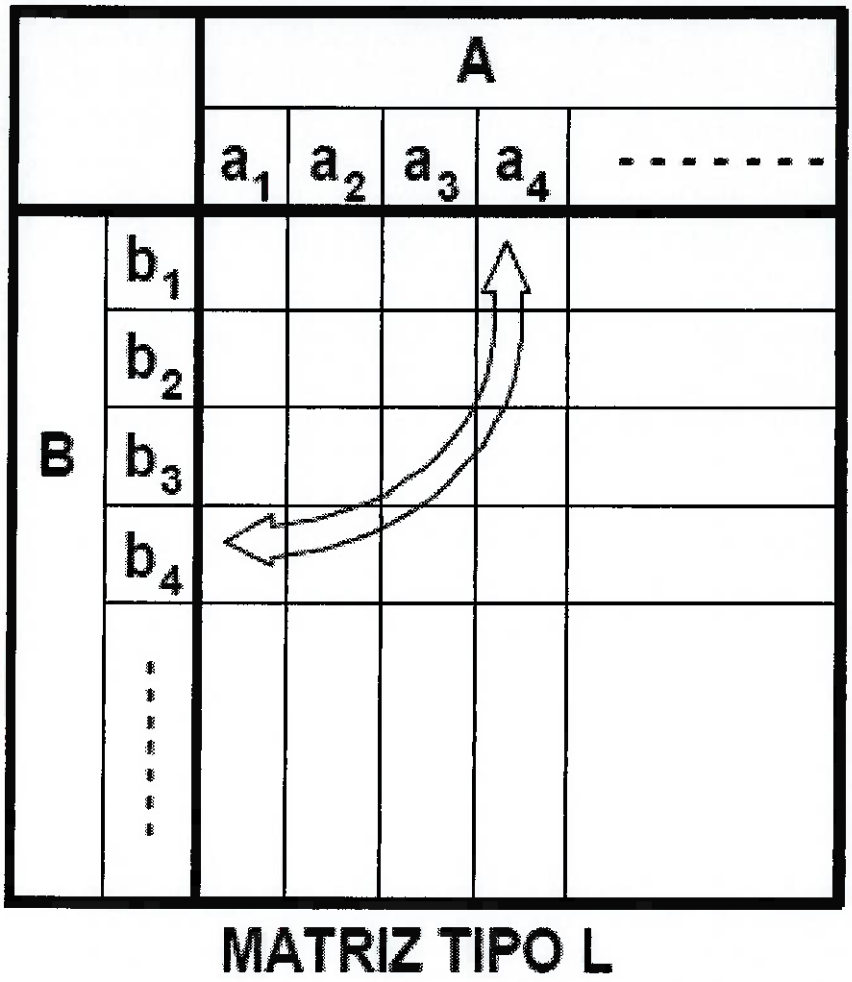

Figura 1: Matriz de priorização

Fonte: Mizuno(1988)

As etapas para elaboração da matriz de priorização podem ser resumidas:

- Definição dos itens
- Definição dos critérios
- Definição dos pesos que serão atribuídos aos critérios
- Avaliar cada item contra cada critério
- Aplicar os pesos nos critérios fazendo as devidas operações matemáticas (Multiplicação)
- Devem ser somados os produtos encontrados em cada coluna
- Reordenar os itens

Um dos requisitos necessários para que a aplicação da ferramenta de matriz de priorização seja efetiva é uma definição de critérios apropriados e atribuição de peso a cada critério de acordo com o grau de impacto que aquele critério provocará no seqüenciamento de itens. Para tal, é aconselhável um profissional com conhecimento aprofundado do processo.

Conforme HEGEDUS (2003) a utilização da matriz de priorização é possível nas seguintes situações:

- Analisar processos de produção onde fatores estão interligados de forma complexa;
- Analisar causa de não conformidades que envolvem grande número de dados;
- Compreender o nível de qualidade desejada, indicado pelos resultado de pesquisas de mercado;
- Classificar, de forma sistemática, as características de percepção da qualidade;
- Acompanhar complexas avaliações de qualidade;
- Analisar dados complexos;

Para este estudo que resultou na elaboração desta monografia, a matriz de priorização foi utilizada para fazer o seqüenciamento de itens (Não-Conformidades) levando em consideração critérios definidos em uma sessão de ***Brainstorming,*** da qual participaram profissionais da Área Técnica que possuíam conhecimento aprofundado do processo.Mais detalhes serão fornecidos na seção **3.3.2 Matriz de Priorização.**

2.1.3 **Diagrama de Causa e Efeito(Ishikawa)**

Conforme NEMESIO (2008) o diagrama de Causa e Efeito (ou Espinha de Peixe) é uma técnica visual que interliga os resultados(efeitos) com os fatores(causas) e é utilizado para identificar , classificar, explorar, ressaltar e analisar visualmente as possíveis causas de um problema.

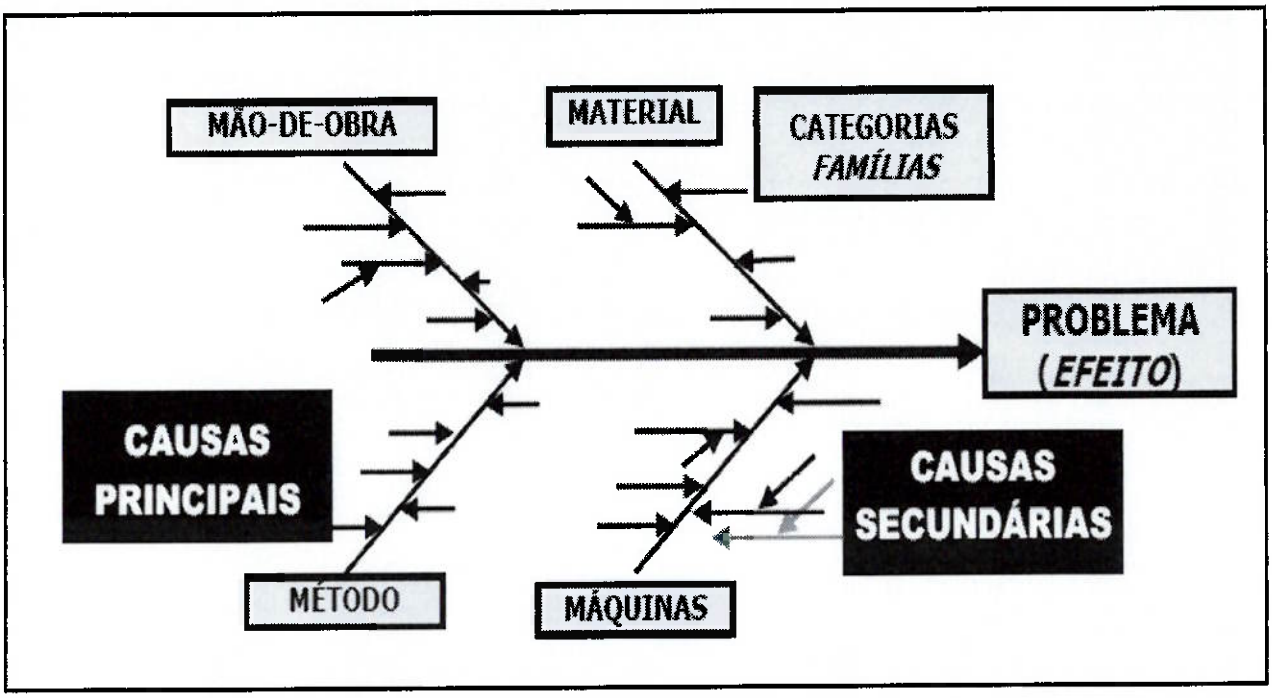

Figura 2: Diagrama de Causa e Efeito

Fonte: NEMESIO(2008)

Conforme ROTANDARO (2002) os passos necessários para elaboração de um Diagrama de Causa e Efeito são:

- Determine um efeito, para tal pode ser utilizada outra ferramenta da Qualidade, no caso deste estudo foi utilizada a Matriz de priorização para definição do Efeito
- Defina os fatores principais: Mao de Obra, Materiais, Máquinas, Métodos, Meio Ambiente e Medição.
- Em cada um dos ramos principais, listar as possíveis causas a eles relacionadas.

Para o estudo realizado nesta monografia a ferramenta Diagrama de Causa e Efeito foi utilizada para mapear as causas da não- conformidade Erro de Lógica de programação(Efeito), mais detalhes serão apresentados na seção **3.3.3 Diagrama de Causa e Efeito(Ishikawa).**

2.1.4 Ciclo PDCA

Segundo Marshall Junior et al (2006), o ciclo PDCA é um método gerencial para a promoção da melhoria contínua e reflete, em suas quatro fases, a base da filosofia do melhoramento contínuo. Por isso, é fundamental que estas fases sejam consecutivas, gerando a melhoria contínua distribuída na organização, estabelecendo a unificação de práticas.

Ainda conforme Marshall Junior et al (2006), apresenta fases do ciclo PDCA, da seguinte forma:

1ª Fase – Plan (Planejamento). Nesta fase é fundamental definir os objetivos e as metas que pretende alcançar. Para isso, as metas do planejamento estratégico precisam ser delineadas em outros planos que simulam as condições do cliente e padrão de produtos, serviços ou processos. Dessa forma, as metas serão só alcançadas por meio das metodologias que contemplam as práticas e os processos.

2ª Fase – Do (Execução). Esta tem por objetivo a prática, por esta razão, é imprescindível oferecer treinamentos na perspectiva de viabilizar o cumprimento dos procedimentos aplicados na fase anterior. No decorrer desta fase precisam-se colher informações que serão aproveitadas na seguinte fase, exceto para aqueles colaboradores que já vêm acompanhando o planejamento e o treinamento na organização.

3ª Fase – Check (Verificação). Fase, no qual é feita a averiguação do que foi planejado mediante as metas estabelecidas e dos resultados alcançados. Sendo assim, o parecer deve ser fundamentado em acontecimentos e informações e não em sugestões ou percepções.

4ª Fase – Act (Ação). A última etapa proporciona duas opções a ser seguida, a primeira baseia-se em diagnosticar qual é a causa raiz do problema bem como a finalidade de prevenir à reprodução dos resultados não esperados, caso, as metas planejadas anteriormente não forem atingidas. Já a segunda opção segue como modelo o esboço da primeira, mas com um diferencial se as metas estabelecidas foram alcançadas.

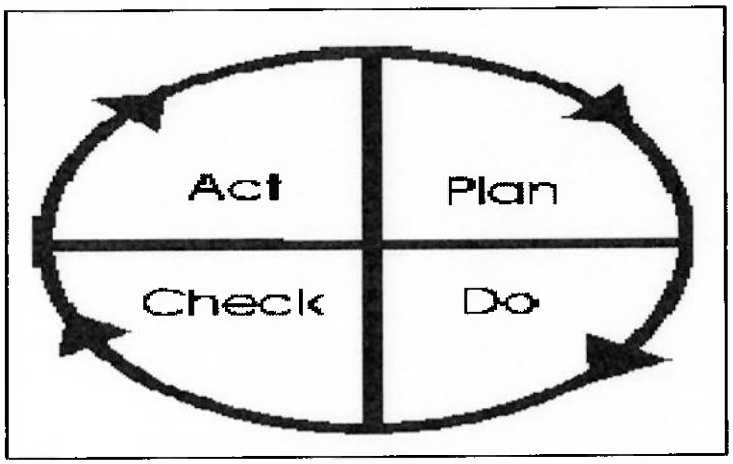

Figura 3: Ciclo PDCA
Fonte: Nemésio(2008)

Para a Instituição Financeira X que é o foco deste estudo, o ciclo PDCA será utilizado juntamente com as ferramentas da Qualidade com o intuito de promover uma melhora no processo de elaboração de software, nas seções posteriores **3.3.2 Matriz de Priorização, 3.3.3 Diagrama de Causa e Efeito(Ishikawa)** e **3.3.4 Plano de Ação** serão feitas análises de como cada uma das ferramentas da Qualidade assim como o Plano de Ação se enquadram no ciclo PDCA.

2.1.5 Erros na Elaboração de softwares

Erro de Lógica de Programação: Erros de Lógica de programação são erros que inviabilizam que o programa faça o que deveria fazer, apesar do código ter sido compilado e executado sem erros. Exemplo: cálculo incorreto, utilização de taxa incorreta, não preenchimento de campo, utilização de tabela incorreta.

Requisito de Negócio não atendido: Os requisitos de Negócio do software devem ser elencados no DEP (Documento de Especificação de Projeto), se por ventura qualquer um dos requisitos de Negócio listados neste documento não for atendido no software entregue para Homologação será caracterizado uma não- conformidade. Exemplo: Um produto bancário deve ter a taxa de juros sempre superior a 5% para satisfazer um requisito legal, mas o programa criado não atende tal restrição, então temos caracterizada uma não conformidade.

Requisito de Técnico não atendido: Os requisitos Técnicos do software devem ser elencados no DEP (Documento de Especificação de Projeto), se por ventura qualquer um dos requisitos Técnicos listados neste documento não for atendido no software entregue para Homologação será caracterizado uma não- conformidade. Exemplo: Não observância tamanho máximo de tabela de dados.

Indisponibilidade de Ambiente de Homologação: Devido alguma falha operacional ou ajuste pontual, o ambiente de homologação poderá ficar indisponível o que impede que os testes de homologação sejam realizados e por sua vez caracteriza uma não- conformidade. Conforme indicador interno, o tempo máximo de restabelecimento do Ambiente de Homologação é de 30 minutos.

Erro de parametrização de Ambiente: É um erro caracterizado por falha de preparação do Ambiente de Homologação para realização dos testes da Homologação. Por exemplo, uma carga invertida de tabelas, ajuste incorreto de parâmetros, etc.

3 ESTUDO DE CASO

3.1 CARACTERIZAÇÃO DA EMPRESA

Para fins metodológicos, a empresa foco deste estudo será identificada pelo codinome **Instituição Financeira X,** será preservada a identidade dos funcionários e das áreas envolvidas, assim como toda informação de caráter confidencial.

A **Instituição Financeira X,** tem sua matriz localizada na cidade de São Paulo, atua no segmento financeiro, a empresa possui mais de 5 mil funcionários o que nos permite categorizá-la como empresa de grande porte.

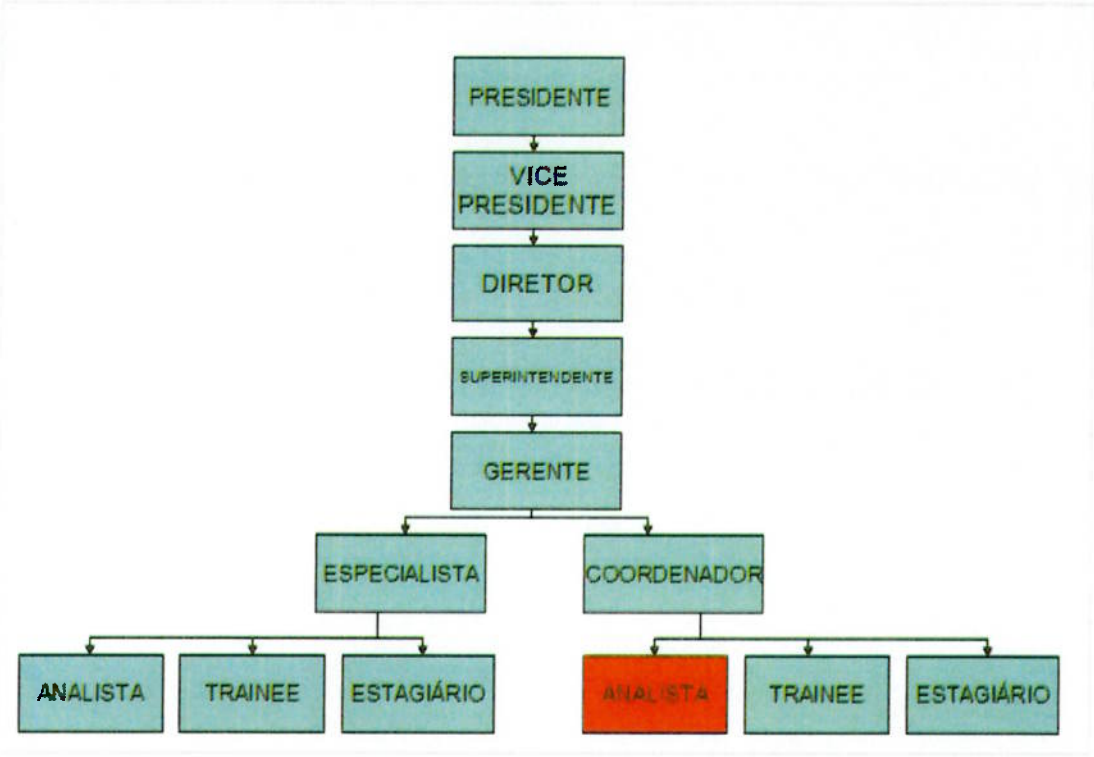

Figura 4: Estrutura organizacional
Fonte: Levantamento em Campo

Cabe ressaltar que a estrutura organizacional mostrada acima, se considerarmos do cargo de Vice-Presidente para baixo a estrutura é replicada para cada segmento de atuação da empresa.

Os profissionais de Área Técnica citados neste trabalho que podem exercer a função de programador, arquiteto de dados, analista de sistemas são identificados na estrutura organizacional e em carteira de trabalho como Analistas.

3.2 PROCESSO DE ELABORAÇÃO DE SOFTWARES

A elaboração do código é feita por um profissional da Área Técnica que poderá ser um Analista Júnior, Analista Pleno ou Analista Sênior, o cargo de Analista Júnior é ocupado por um profissional de pouca experiência, usualmente recém-contratado, com conhecimento técnico mais limitado. O cargo de Analista Pleno é ocupado por um profissional um pouco mais experiente, com cerca de 3 anos de atuação na área, com conhecimento técnico mais abrangente. E por fim o cargo de Analista Sênior é ocupado por um profissional muito experiente, normalmente com 6 anos ou mais anos de atuação na área, com conhecimento técnico bem amplo.

O processo de elaboração de software pode sofrer pequenas variações de acordo com diretrizes técnicas e/ou operacionais de cada empresa, mas de um modo geral o processo pode ser subdividido conforme figura abaixo:

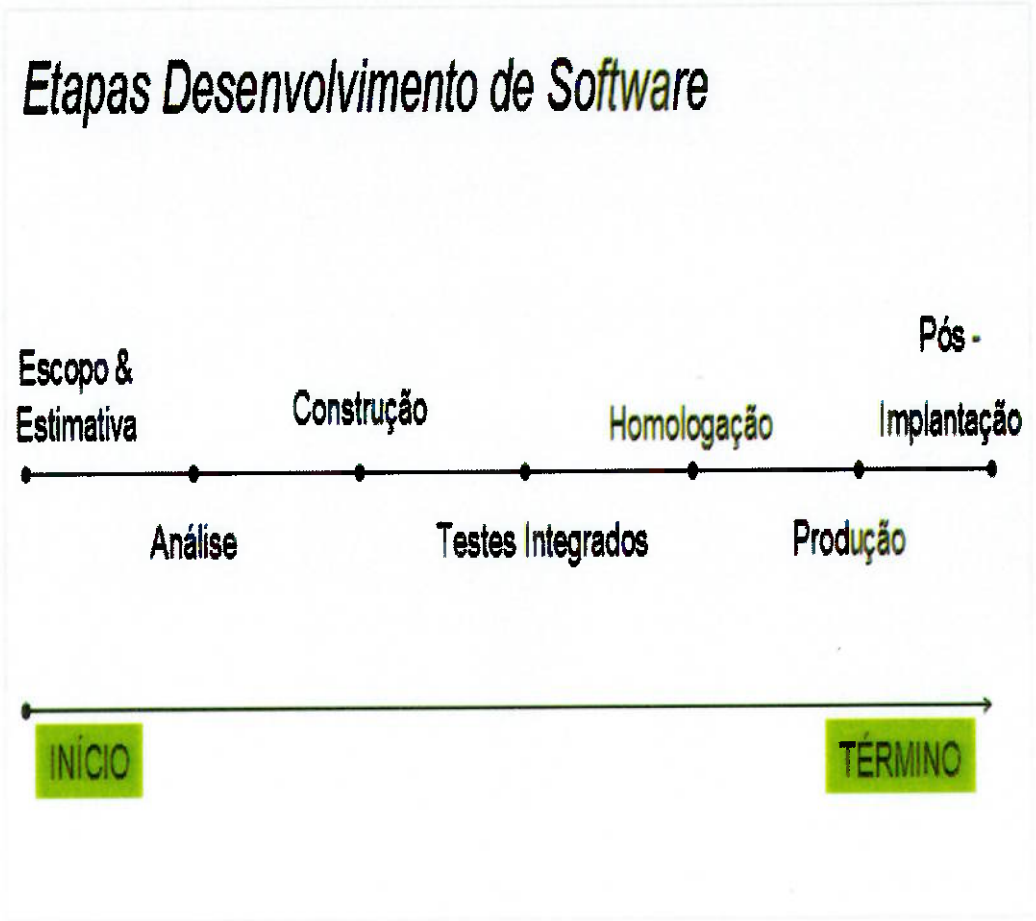

Figura 5: Etapas Elaboração de Software
Fonte: Levantamento em Campo

Importante ressaltar que a duração de cada etapa e do desenvolvimento como um todo pode variar de acordo com o grau de complexidade do projeto, quantidade de recursos financeiros, materiais e humanos e o conjunto de todas as etapas é conhecido como **ciclo de vida do projeto**[4].

[4] **Ciclo de vida do projeto**= Ciclo de vida de um projeto define as fases que interligam o começo do projeto ao seu final

A fase de **Escopo & Estimativa** é o momento onde deve ser realizada a descrição do escopo do software, devem-se também ser estabelecidos os limites de projeto, apresentação da lista de restrições e premissas de projeto.

Na fase de **Análise** a equipe técnica verifica a viabilidade de execução do projeto.

Na fase de **Construção** os programadores devem criar o código do software utilizando uma linguagem de programação, as últimas dúvidas das premissas e requisitos solicitados pela área solicitante devem ser dirimidas nesta fase para evitar retrabalho futuro.

A fase de **Testes Integrados** é executada pelos programadores, cabe ressaltar que é uma prática comum que sejam executados os testes unitários primeiramente, nos testes unitários é testado individualmente cada funcionalidade, cálculo, interface com o usuário. Já nos Testes Integrados é testado o programa como um todo.

Com o término da fase de Testes Integrados o software é liberado para área solicitante para que seja feita a **Homologação** que nada mais é do que a validação do programa, que é feita normalmente utilizando **casos de teste** que retratam situações reais.

A fase de Homologação termina com o aceite formal da área solicitante, reconhecendo que o software entregue já está apto a ser utilizado, a equipe técnica faz toda parametrização de ambiente necessária (Carga de Tabelas, equalizações, etc.) e o software é implantado, este processo é conhecido como entrada na etapa de **produção.**

Com o software em produção, ainda pode eventualmente ser necessário algum ajuste pontual e/ou correção emergencial, tal etapa é conhecida como **Pós-Implantação.**

Para facilitar o entendimento do processo de elaboração de softwares, interfaces do processo, áreas envolvidas e possíveis pontos críticos do processo será utilizado o mapa de processo.

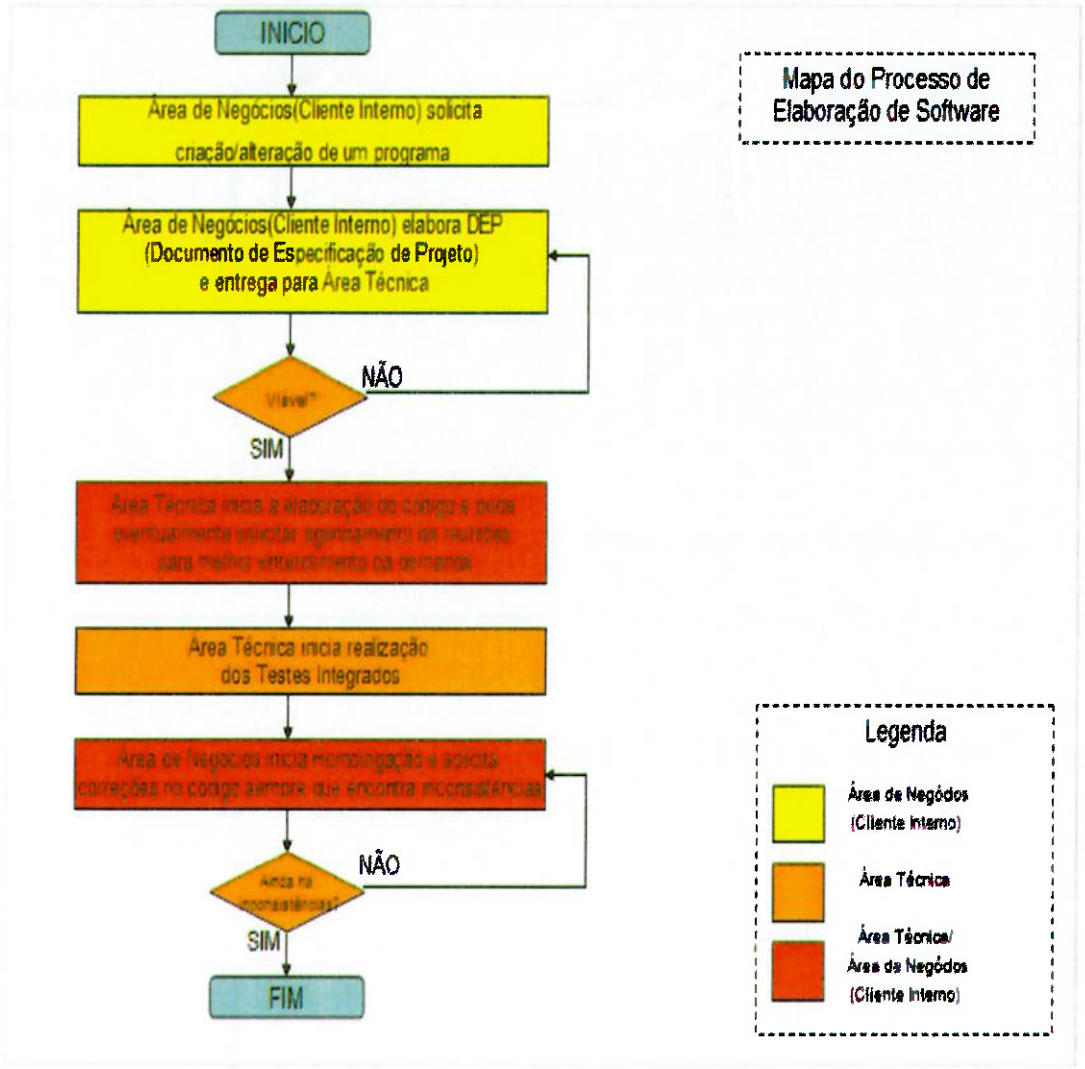

Figura 6: Mapa do Processo de Elaboração de Software
Fonte: Levantamento em Campo

O processo se inicia com a identificação por parte da Área de Negócios (Cliente Interno) da necessidade de criação ou alteração de um programa, a criação de um programa pode ter sido solicitada com o intuito de oferecer um novo produto (Serviço bancário) aos clientes, já a alteração pode ter sido solicitada devido a mudanças de requisitos legais (Alteração de tributação de um produto bancário, descontinuidade de um produto, etc.).

A Área de Negócios elabora o DEP (Documento de Especificação de Projeto), documento que deverá conter os requisitos (Técnicos, Processuais e de Negócios) necessários para atender a demanda e será o documento a partir do qual a Área Técnica avaliará a viabilidade técnica ou não de execução do projeto, tal avaliação é formalizada com a emissão de um parecer.

Caso o parecer não seja favorável, o documento é devolvido para a Área de Negócios (Cliente Interno) que poderá optar por desistir da demanda ou reestruturar o DEP com alteração dos requisitos que não puderam ser atendidos e submeter o documento novamente para análise da Área Técnica.

Caso o parecer seja favorável, a Área Técnica iniciará o processo de elaboração do código e neste período serão realizadas reuniões periódicas de elicitação de requisitos com a Área de Negócios (Cliente Interno) para haja maior entendimento possível da demanda e para assegurar que o código que está sendo elaborado conforme necessidade da Área de Negócios (Cliente Interno).

A fase de Testes Integrados é realizada pelo Analista que elaborou o código, nesta fase serão realizados testes unitários, testes de desempenho e outros tipos de testes que visam reduzir a quantidade de erros no código para que a próxima etapa de Homologação seja realizada da melhor forma possível.

Na fase de Homologação a Área de Negócios testa o programa entregue, e a cada inconsistência identificada eles contatam a Área Técnica e solicita correção, no momento que não houver mais inconsistências a Área de Negócios formaliza a aceitação do programa com um documento de "ACEITE".

3.3 APLICAÇÃO DAS FERRAMENTAS DE QUALIDADE

3.3.1 Levantamento de Dados

Para aplicação das ferramentas da Qualidade e do ciclo PDCA, foi realizado um **Brainstorming** com o intuito de definir a melhor forma de levantamento de dados, nesta reunião participaram integrantes da Área Técnica, desta reunião concluiu-se que a melhor forma de levantamento de dados seria através do envio de um formulário para o ponto focal da Área de Negócios (Cliente Interno) que eventualmente poderia ser um gerente, coordenador de projetos ou Analista Sênior.

Este formulário deveria ser preenchido com recorrência quinzenal até que houvesse o término da fase de Homologação. Foi decidido também no **Brainstorming** que as opções de não-conformidades seriam categorizadas, de modo que o ponto focal marcaria no formulário opções pré-definidas.

O formulário também possuía uma seção que não deveria ser preenchida pelo ponto focal da Área de Negócios (Cliente Interno), esta seção deveria ser preenchida pela Área Técnica assim que recebesse o formulário preenchido. O responsável da Área Técnica tinha a incumbência de analisar o grau de complexidade do software (Baixa complexidade, Média Complexidade e Alta Complexidade) para tal classificação seria utilizado o parâmetro:

- Baixa complexidade até 50 linhas de código
- Média complexidade maior que 50 linhas de código e menor que 200.
- Alta complexidade mais do que 200 linhas de código

Tal classificação foi feita com o intuito de delimitar o escopo deste trabalho, tendo em vista o caráter experimental deste trabalho, que aplicará as ferramentas e conceitos da qualidade unicamente para os softwares de baixa complexidade.

Lembrando que para estabelecimento do grau de complexidade de um software há técnicas mais apuradas. Segundo McCabe (1976) que desenvolveu a estratégia ***Basis Path Testing***, que nada mais é do que uma métrica para apurar a complexidade de um software e que consiste em testar cada caminho linearmente independente do código fonte, neste caso quanto maior a quantidade de caminhos maior a complexidade.

Mas dado o objetivo deste trabalho, o esforço que seria necessário para aplicação de tal método não seria justificável.

3.3.2 Matriz de Priorização

Com o recolhimento dos formulários enviados a Área Técnica utilizou a ferramenta de qualidade Matriz de Priorização, conforme informado na seção introdutória, tal ferramenta aplica-se na priorização ou estabelecimento de seqüências de itens com base em um ou mais critérios.

Para o estabelecimento dos critérios levou-se em consideração que a **Instituição financeira X** tem como prioridade operacional neste momento a aplicação dos conceitos do PMBOK, documento que relata as boas práticas de projeto, e conforme o PMBOK na gestão de projeto deve haver observância da ***triple constraint*** [5], *a* restrição tempo é observada no critério tempo médio de correção, a restrição custo é observada no critério custo médio de correção e a restrição qualidade é observada no critérios quantidade de não conformidades e gravidade das não conformidades.

---

[5] **Triple Constraint**= Restrição tripla, conceito utilizado em gerenciamento de projeto, sendo que as três restrições são tempo, custo e escopo.

Tal ferramenta também exige o estabelecimento de pesos para os critérios, pois os critérios podem eventualmente possuir diferentes impactos no seqüenciamento de itens. Os pesos para cada critério foram determinados com a realização de um ***Brainstorming*** por profissionais da Área Técnica e concluiu-se que o critério Quantidade, Custo Médio de correção e Tempo Médio de correção tinham peso 3 e o critério Gravidade peso 2.

As informações necessárias para o preenchimento da matriz referente a quantidade de não conformidades e gravidade de não conformidades foram extraídas dos formulários preenchidos pela Área de Negócios(Cliente Interno), já as informações Tempo e Custo médio por correção foram obtidas por Brainstorming realizado por profissionais da Área Técnica.

Todas as informações utilizadas no preenchimento da tabela foram ajustadas para ficarem no intervalo de 0 a 10, por exemplo no caso de quantidade de não conformidades, o caso mais crítico era da não conformidade **Erro de lógica de programação**[6] que tinha valor 301, deste modo foi atribuído a este item o valor 10 e para as demais não conformidades foi aplicado regra de três simples para manter devida proporcionalidade.

---

[6] **Erro de Lógica de Programação**= São erros ocorridos na elaboraçao do código, e usualmente ocorrem por falta de alinhamento(falha de comunicaçao) do programador e do usuário final.

A matriz de priorização deste estudo de caso já com a aplicação dos pesos e ponderações citados:

	Critérios				
Não Conformidades	Quantidade	Gravidade	Custo Médio Correção	Tempo Médio de correção(Dias)	TOTAL
Erro de Lógica Programação	30,000	13,646	30	27	100,646
Erro Requisito de Negócio não atendido	23,820	14,704	24	30	92,524
Erro Requisito de Técnico não atendido	10,863	12,352	21	12	56,215
Indisponibilidade Ambiente de Homologação	2,691	3,058	3	0,006	8,755
Erro parametrização de Ambiente	0,807	9,176	12	6	27,983

Tabela 3: Matriz de Priorização Estudo de Caso
Fonte: Levantamento em Campo

O seqüenciamento de itens conforme a matriz apresentada será:

1° Erro de Lógica de Programação
2° Erro Requisito de Negócio não atendido
3° Erro Requisito Técnico não atendido
4° Erro parametrização de Ambiente
5° Indisponibilidade Ambiente de Homologação

Tabela 4: Seqüenciamento de Itens
Fonte: Levantamento em Campo

Podemos observar que a utilização da Matriz de priorização nos permitiu uma análise mais pormenorizada, pois se estivéssemos só nos baseando em um único critério o resultado poderia ter sido outro, se tomássemos unicamente, por exemplo,

o critério gravidade a não conformidade Erro Requisito de Negócio não atendido seria priorizada por ter maior pontuação, já com a utilização da Matriz de priorização conseguimos fazer uma análise com vários critérios.

Segundo SEACORD e PLAKOSH (2003) o termo correção de software é muito amplo e se refere a quaisquer mudanças feitas nos softwares após eles terem sido entregues para a área solicitante, mas ainda assim é possível delimitar quase que a totalidade das correções em 4 categorias:

- Corretiva: São usualmente decorrentes de erro de lógica de programação. Estas correções são feitas para corrigir erros no sistema. Os erros que fazem com que o software tenha comportamento inconsistente mesmo tendo sido observado os requisitos da especificação de projeto. A atividade corretiva normalmente ocorre quando um erro é reportado por um usuário final no momento que o mesmo observa comportamento inesperado do software.

- Adaptativa: Estas correções são feitas para manter os softwares atualizados frente a mudanças constantes, por exemplo novos sistemas operacionais, compiladores de linguagem de programação, etc.

- Aprimoramento: Que são correções para aprimorar o produto, por exemplo a inserção de novos requisitos funcionais, ou aprimoramento de performance ou ainda outros atributos de sistema.

- Preventiva: Estas correções são feitas para prevenir futuras correções e dar maior credibilidade ao software. Ao contrário das 3 categorias anteriormente citadas para correção, neste caso as correções buscam de modo pró ativo simplificar a evolução natural do software.

Os autores ainda reportam que estudos comprovam que cerca de 90% das correções se enquadram nas categorias corretiva e adaptativa. Não obstante, estudos complementares mostram que dependendo do tipo de software as correções corretivas por si só representam mais de 70% de todas as correções efetuadas.

Os dados obtidos nos levantamentos realizados nesta monografia juntamente com a aplicação do ***Brainstorming*** e da ferramenta de qualidade Matriz de Priorização corroboram o estudo feito pelos autores, tendo em vista que o erro mais crítico, de maior severidade foi o erro de lógica de programação que se enquadra na categoria corretiva, deste modo foi observada uma consonância do trabalho aqui realizado com o estudo publicado dos autores.

Com a identificação da não conformidade mais crítica é possível dar continuidade ao trabalho.

3.3.3 Diagrama Causa e Efeito(Ishikawa)

Tendo em vista que foi possível determinarmos o **Efeito** através da utilização da Matriz de priorização, que conforme observado no tópico **3.3.1 Matriz de Priorização** foi **Erro de Lógica de Programação,** o próximo passo será a aplicação de outra ferramenta da qualidade conhecida por Diagrama de Causa e Efeito que nos permitirá identificar possíveis causas que teriam ocasionado tal efeito.

As causas serão subdivididas em 5 categorias: Mão de Obra, Material, Meio Ambiente, Método e Máquina

A definição das causas foi feita com a realização de um Brainstorming por parte de profissionais da Área Técnica, também foi utilizado como insumo o resultado consolidado do **termômetro**[7].

---

7 Termômetro= pesquisa de clima organizacional que é realizada com periodicidade mensal e nela os funcionários, de modo anônimo, relatam os principais inconvenientes e/ou dificuldades que tem encontrado no ambiente de trabalho para exercer tuas atividades.

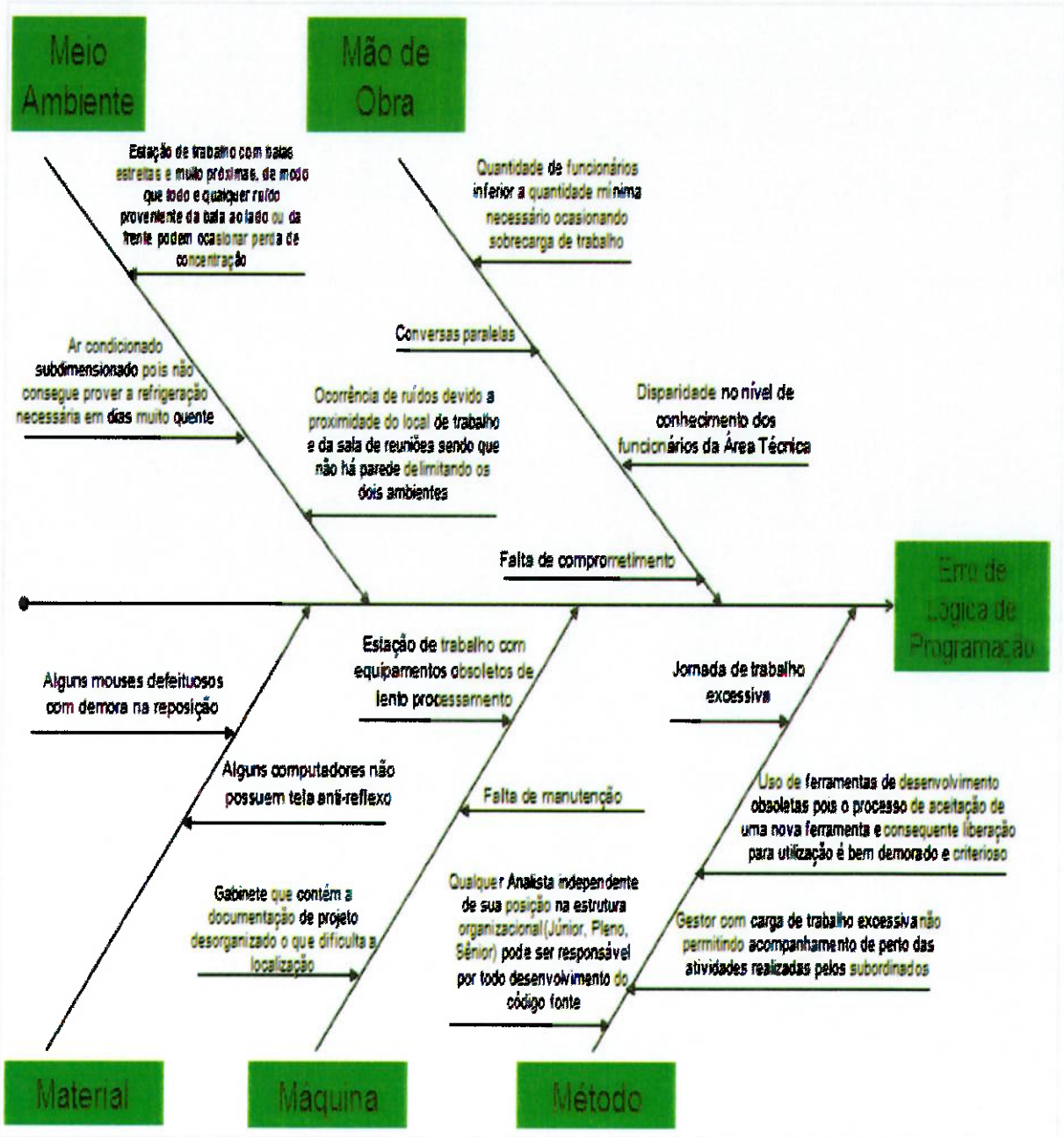

Figura 7: Diagrama de Causa e Efeito do Estudo de Caso
Fonte: Levantamento em Campo

3.3.4 Plano de Ação

Dentre as causas listadas com a utilização da ferramenta Diagrama de Causa e Efeito é necessário que seja feita uma escolha de quais causas poderão ser abordadas neste momento, esta análise envolve restrições financeiras, técnicas, logísticas.

Conforme PÁDUA e MENDES (2009) a busca da qualidade no desenvolvimento de software é um processo sistemático que deve focalizar todas as etapas e artefatos produzidos com o objetivo de garantir a conformidade de processos e produtos, prevenindo e eliminando defeitos. No desenvolvimento de software, a qualidade do produto está diretamente relacionada à qualidade do processo de desenvolvimento. Desta forma, é comum que a busca por um software de maior qualidade passe necessariamente por uma melhoria no processo de desenvolvimento.

Segundo MYERS (1979) o custo de correção de erros cresce exponencialmente à medida que o processo de desenvolvimento avança, se tomarmos X= custo de correção na fase de especificação, segundo o autor se o erro for identificado na fase de testes o custo será 10X e caso este erro só venha a ser identificado na fase de produção o custo será de 100X.

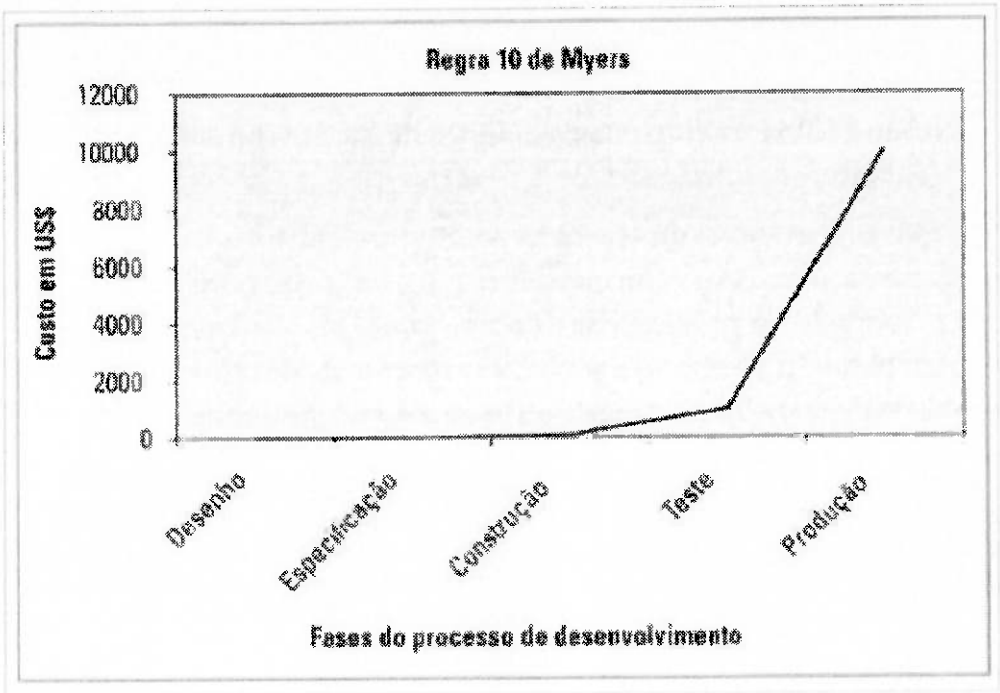

Figura 8: Regra 10 de Myers
Fonte: Myers(1979)

Conforme foi visto no item 3.2 PROCESSO DE ELABORAÇÃO DE SOFTWARES a elaboração do código do software pode ser feita por um programador júnior, pleno ou sênior, os quais possuem diferentes níveis de conhecimento e evidentemente tal discrepância no nível de conhecimento pode interferir na qualidade final do código elaborado, sendo importante ressaltar que do modo que o processo está estruturado hoje em dia, um programador júnior poderá elaborar todo o código por conta própria.

Deste modo após a realização de um ***Brainstorming*** o primeiro plano de ação escolhido foi a criação de uma etapa adicional de análise de integridade do código elaborado e tal atividade deverá necessariamente ser realizada por um analista sênior. Ou seja, todo software elaborado por um analista júnior, pleno e até mesmo sênior deverá ser submetido à análise de integridade por outro analista sênior antes de ser liberado para a Área Cliente homologar. Deste modo se analisarmos 3.3.3 Diagrama Causa e Efeito(Ishikawa) a causa escolhida para aplicação do plano de ação foi "Disparidade no nível de conhecimento dos Funcionários da Área Técnica".

Na etapa adicional de análise de integridade um analista sênior deverá analisar o código elaborado com o intuito de identificar possíveis inconsistências e quando as identificar deverá corrigí-las, a inserção de tal etapa visa promover uma melhoria no processo de desenvolvimento e identificar precocemente erros que só seriam identificados na etapa posterior de Homologação.

Dado que havia apoio da alta direção foi designado um analista sênior para dedicação exclusiva à fase de análise de integridade, sendo que na impossibilidade deste recurso realizar tal atividade, outro analista de mesmo nível de conhecimento(Sênior) ou gradação imediatamente inferior(Pleno) deveria assumir. Foi ainda definido o **SLA**(Service Level Agreement) de 1 dia, o SLA nada mais é, neste contexto, o tempo que será necessário para a realização da etapa de análise de integridade.

Abaixo está a figura que localiza a etapa de analise de Integridade ao longo do eixo temporal que contem as etapas de desenvolvimento de software:

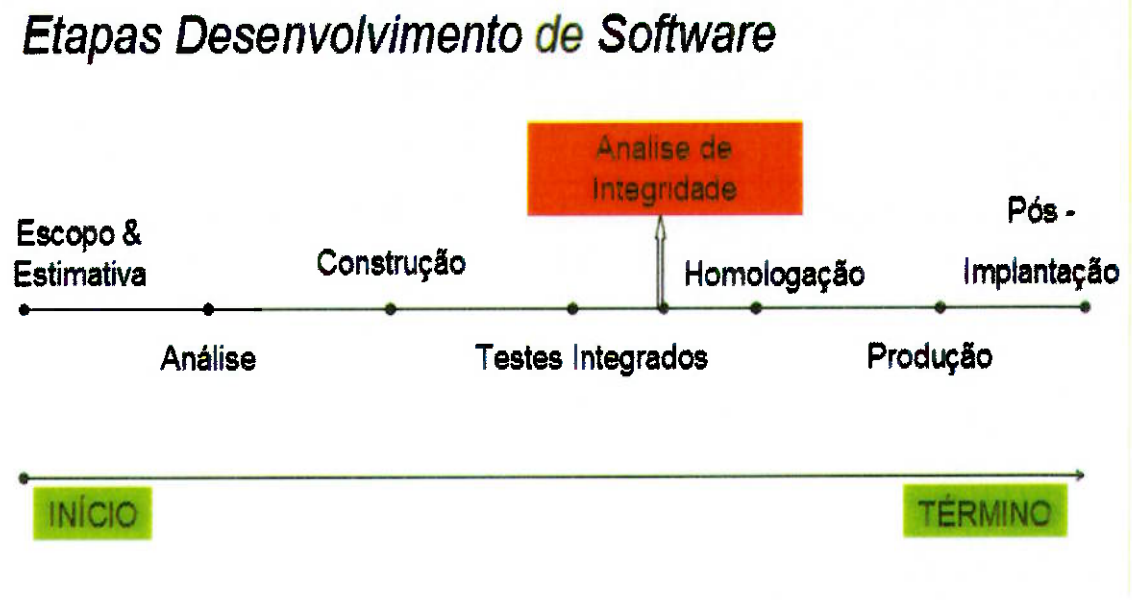

Figura 9: Analise de Integridade inserida nas Etapas de Desenvolvimento de Software
Fonte: Levantamento em Campo

Se verificarmos 3.3.3 Diagrama Causa e Efeito(Ishikawa) temos a causa "Alguns mouses defeituosos com demora na reposição", também foi estabelecido um plano de ação para esta causa que consistiu na compra imediata de 30 mouses para deixá-los em estoque e permitir rápida reposição quando necessário. Este plano de ação provavelmente não será o que mais auxiliará para obtenção de melhor qualidade final no software entregue, mas por sua vez é um plano de ação de fácil exeqüibilidade e complementar, a demora na reposição de mouses defeituosos provocava insatisfação nos funcionários por terem problemas com falta de ferramenta de trabalho e até mesmo interferiam na produtividade, já que enquanto não tinha as devidas ferramentas de trabalho o funcionário ficava ocioso.

4 RESULTADO E DISCUSSÕES

4.1 APRESENTAÇÃO DOS RESULTADOS

Com o primeiro recolhimento dos formulários enviados tivemos a seguinte quantidade de não- conformidades (Erro de Lógica de Programação, Erro requisito de Negócio não atendido, Requisito Técnico não atendido, Erro indisponibilidade ambiente de homologação e Erro parametrização ambiente).

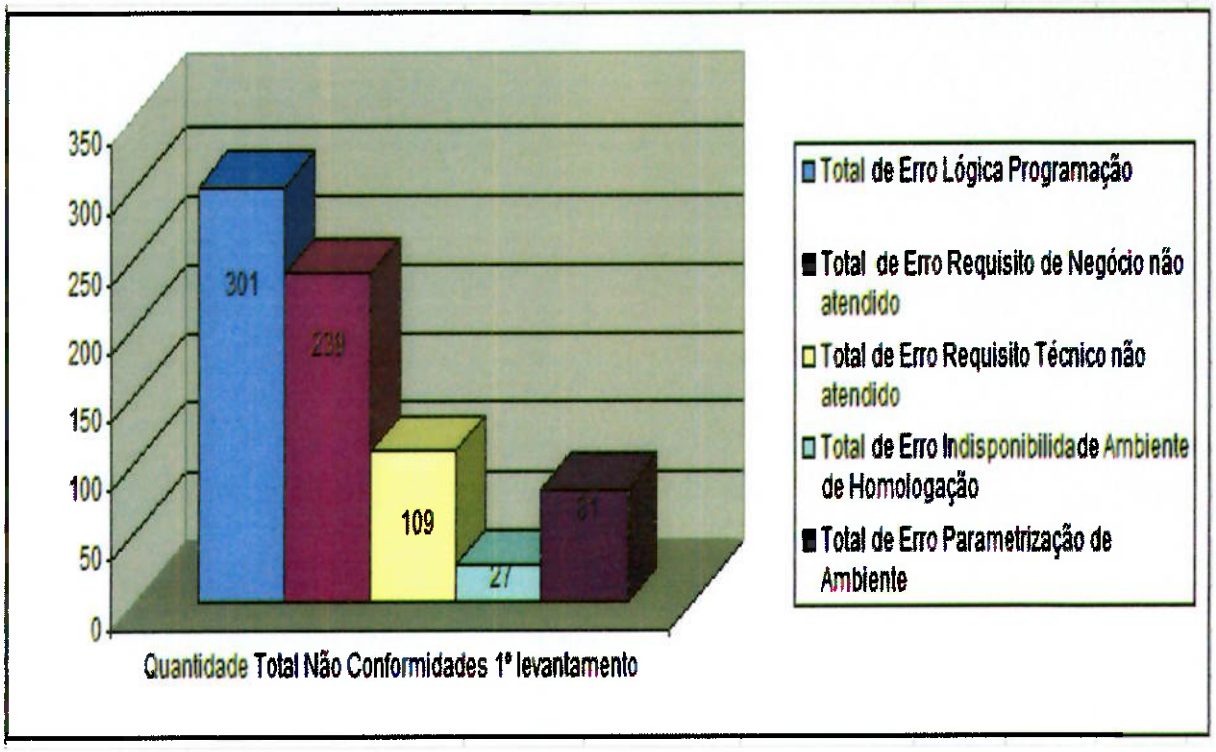

Figura 10: Primeiro levantamento de não- conformidades

Fonte: Levantamento em Campo

Conforme explicado no item **3.3.1 Levantamento de Dados** neste trabalho só foram abordados os softwares produzidos que se enquadram na categoria de baixa complexidade, de modo que no primeiro levantamento houve 17 softwares com tal classificação. O que nos permitiu inferir :

- Média Erro de Lógica Programação = 17,71
- Média Erro Requisito de Negócio não atendido = 14,06
- Média Requisito de Técnico não atendido = 6,41
- Média Total Indisponibilidade Ambiente de Homologação= 1,59
- Média Total Erro parametrização de Ambiente = 4,76

No período imediatamente posterior ao primeiro levantamento e apuração de dados juntamente com o apoio da alta direção iniciou-se a implementação do plano de ação discutido no item **3.3.4 Plano de Ação.**

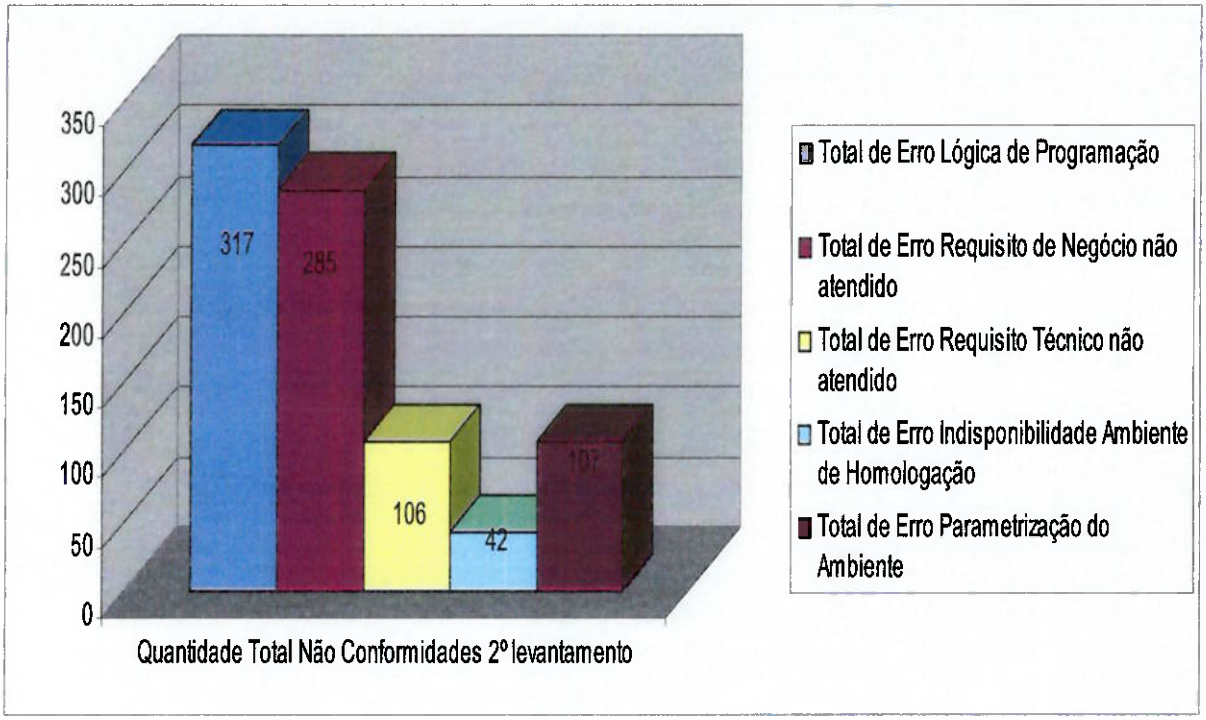

Figura 11: Segundo levantamento de não- conformidades

Fonte: Levantamento em Campo

No segundo levantamento houve 19 softwares que se enquadraram na categoria almejada. O que nos permitiu inferir:

- Média Erro de Lógica Programação = 16,68
- Média Erro Requisito de Negócio não atendido = 15,05
- Média Requisito de Técnico não atendido = 5,58
- Média Total Indisponibilidade Ambiente de Homologação= 2,21
- Média Total Erro parametrização de Ambiente = 5,63

Foi observado que a media de erros de lógica de programação por software elaborado sofreu uma redução de 17,71 para 16,68 logo na primeira quinzena de

aplicação do plano de ação, o que em termos percentuais representa uma redução de 5,81%.

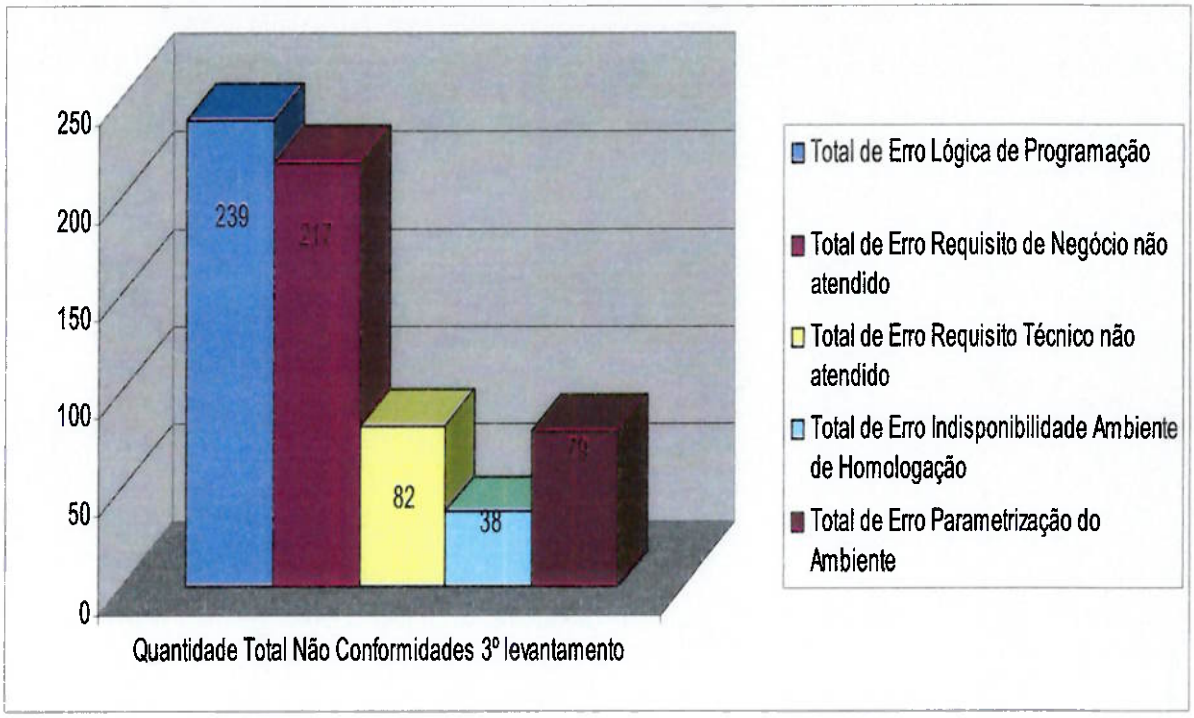

Figura 12: Terceiro levantamento de não- conformidades
Fonte: Levantamento em Campo

No terceiro levantamento houve 15 softwares que se enquadraram na categoria almejada. O que nos permitiu inferir:

- Média Erro de Lógica Programação = 15,93
- Média Erro Requisito de Negócio não atendido = 14,47
- Média Requisito de Técnico não atendido = 5,47
- Média Total Indisponibilidade Ambiente de Homologação= 2,53
- Média Total Erro parametrização de Ambiente = 5,27

Foi observado que a media de erros de lógica de programação por software elaborado sofreu uma redução de 16,68 para 15,93 no intervalo compreendido entre a terceira e a quarta quinzena o que em termos percentuais representa uma redução de 4,49%.

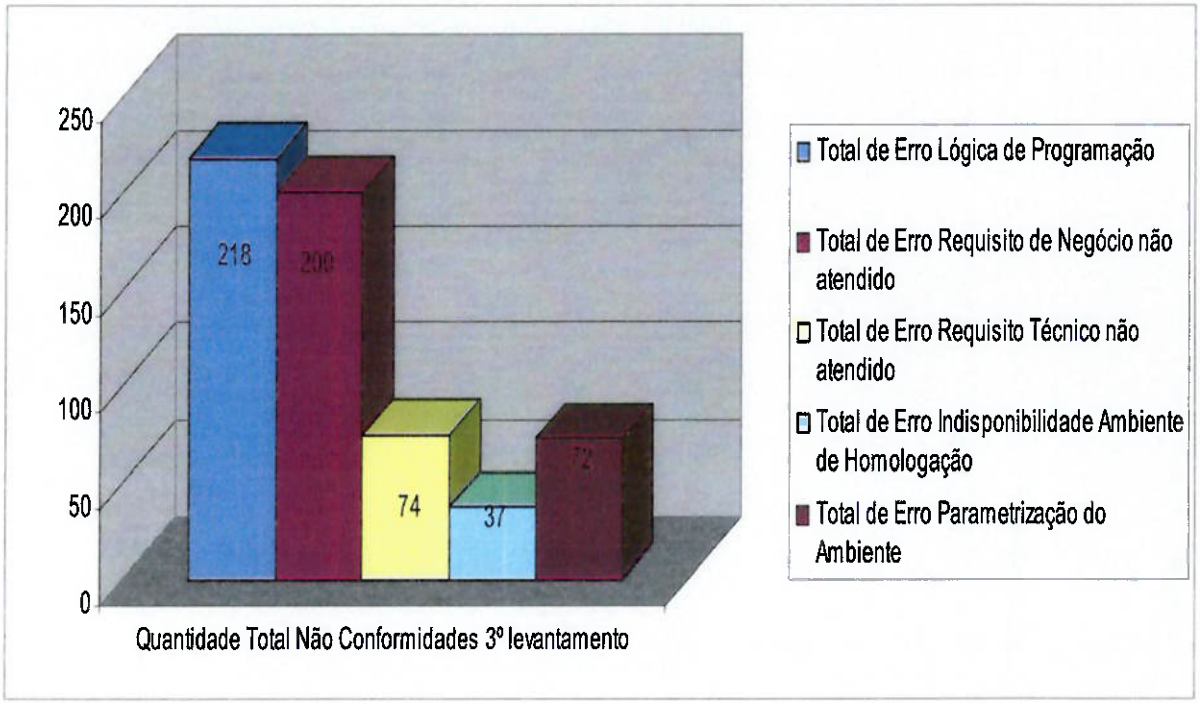

Figura 13: Quarto levantamento de não- conformidades
Fonte: Levantamento em Campo

No quarto levantamento houve 14 softwares que se enquadraram na categoria almejada. O que nos permitiu inferir:

- Média Erro de Lógica Programação = 15,57
- Média Erro Requisito de Negócio não atendido = 14,29
- Média Requisito de Técnico não atendido = 5,29
- Média Total Indisponibilidade Ambiente de Homologação= 2,64
- Média Total Erro parametrização de Ambiente = 5,14

Foi observado que a media de erros de lógica de programação por software elaborado sofreu uma redução de 15,93 para 15,57 durante a quarta quinzena o que em termos percentuais representa uma redução de 2,25%.

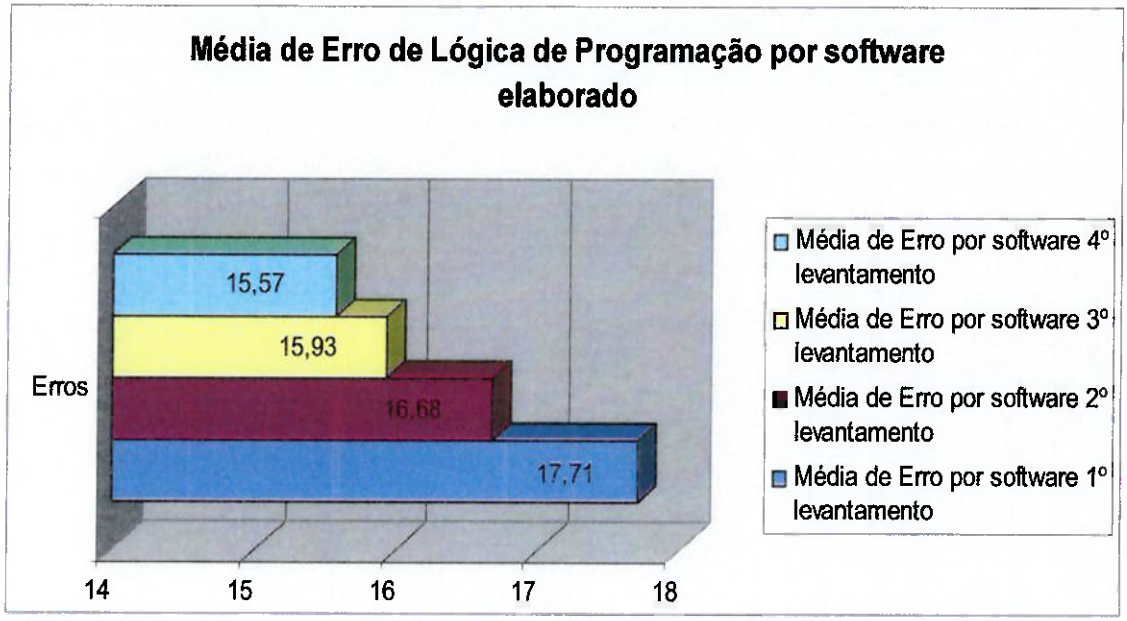

Figura 14: Evolução media de erro de lógica de programação por software elaborado
Fonte: Levantamento em Campo

Com a execução do plano de ação de submeter todos os códigos fontes elaborados a revisão de um analista Sênior antes que os mesmos fossem liberados para Área de Negócios(Cliente Interno), obtivemos uma diminuição da media de erros de lógica de programação por programa de 17,71 para 15,57 o que representa redução de 12,1% na quantidade da não-conformidade "Erro de Lógica de Programação" no intervalo de 2 meses.

Nesta fase se considerarmos o ciclo PDCA estaríamos falando da fase de **check,** onde são verificados se as ações executadas surtiram resultado

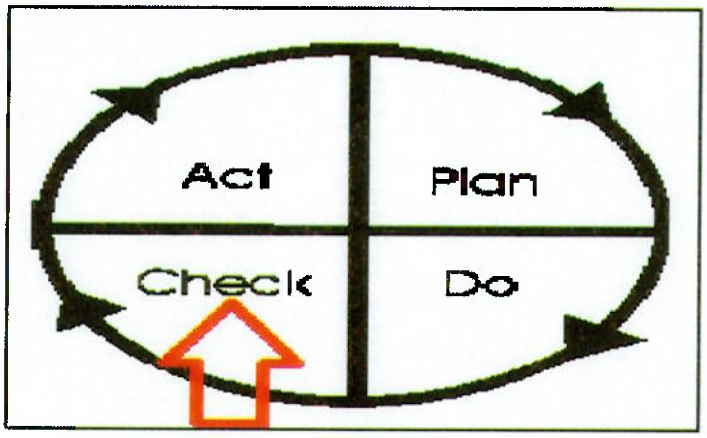

Figura 15: Ciclo PDCA
Fonte: Levantamento em Campo

5 CONCLUSOES

O objetivo proposto deste trabalho era aprimorar o processo de elaboração de softwares da **Instituição Financeira X** com o intuito de diminuir a quantidade de não- conformidades, tal objetivo foi atingido conforme apresentado na seção 4.1 APRESENTAÇÃO DOS RESULTADOS houve uma diminuição na ocorrência da não- conformidade mais crítica "erros de lógica de programação" da ordem de 12,1% por software elaborado.

Porém conforme observado por KIERAS (2009) o desenvolvimento de software é difícil devido:

- Motivos técnicos e não técnicos
- Provavelmente a atividade mais difícil que o ser humano criou pra si mesmo
- Nós não sabemos de fato como fazer bem

Tais observações podem parecer um pouco exageradas, mas é evidente o elevado grau de complexidade do desenvolvimento de software, tendo em vista que são realizados investimentos vultosos no setor de Tecnologia da Informação e ainda sim, conforme exposto na seção 1.3 JUSTIFICATIVA, a indústria de software apresenta grande imaturidade.

Uma das grandes dificuldades observadas neste estudo aqui apresentado consiste em conseguir mensurar a eficácia de um plano de ação aplicado, já que uma não conformidade decorrente do processo de desenvolvimento de software pode ter ocorrido a partir de n possíveis causas. Por exemplo, o fato de ter havido uma diminuição de 12,1% na ocorrência da não conformidade erro de lógica de programação não necessariamente nos permite atribuir tal resultado a ação única do plano de ação que consistiu em submeter o software elaborado a revisão de um analista sênior, pois neste intervalo de tempo de 2 meses onde ocorreu a aplicação do plano de ação, n outras variáveis que poderiam influir em um aumento ou redução da ocorrência desta não conformidade não permaneceram estáticas, por exemplo, neste período houve treinamento/capacitação de programadores o que teoricamente implicaria numa redução da ocorrência da não conformidade citada,

mas por outro lado foi o período de férias de alguns programadores sobrecarregando os programadores que não estavam de férias, esta sobrecarga de trabalho pode eventualmente implicar em um aumento da ocorrência de não conformidades.

Se observamos na seção 4.1 APRESENTAÇÃO DOS RESULTADOS a oscilação da quantidade de ocorrência por software elaborado, tanto para mais quanto para menos, retrata bem a dificuldade citada no parágrafo anterior.

Ainda assim os resultados foram bem recebidos pela alta direção, até porque já foi bem difundido na **Instituição Financeira X** que a aplicação do ciclo PDCA é um processo para obtenção de melhoria contínua e que para tal, não se restringirá unicamente ao resultado isolado de um único plano de ação.

5.1 SUGESTÕES PARA TRABALHOS FUTUROS

Com a aplicação da ferramenta da qualidade Matriz de priorização pudemos observar que as não-conformidades "Erro de Lógica de Programação" e " Erro de Requisito de Negócio não atendido" foram as que apresentaram maior criticidade, para este trabalho identificamos as possíveis causas para a não- conformidade "Erro de Lógica de Programação" aplicando Ishikawa, como sugestão para trabalho futuro seria aplicar a ferramenta Ishikawa para a não-conformidade " Erro de Requisito de Negócio não atendido" pois se trata da segunda não-conformidade de maior significância para o processo e a identificação de suas causas pode permitir ganhos ainda melhores na qualidade do produto final(Software).

Conforme KIERAS (2009) dentre os fatores que implicam nos softwares entregues não possuírem maior qualidade o autor ressalta:

- Freqüente falha de comunicação entre programadores e área de negocio
- Programadores muitas vezes não desenvolvem o que o usuário final quer

Tais observações só vem reforçar o quão importante é a não conformidade "Erro de Requisito de Negocio não atendido", pois a falha de comunicação entre programador e área de negocio pode culminar no fato do programador ter a "percepção" que entendeu o requisito de negócio no DEP, mas no fundo pode ter interpretado incorretamente o documento e isto implicará na elaboração de um código de programa com erro de requisito de negocio não atendido, o qual por sua vez provavelmente só será identificado pela área de negocio na etapa de homologação.

REFERENCIAS BIBLIOGRÁFICAS

CASSIOLATO, J. E. **A Conexão entre Usuários e Produtores de Alta Tecnologia: Um Estudo de Caso**. *Ensaios FEE*, v .13, nº 1, 1992

DIEESE . **A Terceirização no Setor Bancário**. Boletim do DIEESE. São Paulo, v.8 , 1997.

FORBES. **Índice Forbes Global 2000.** 2009. Disponível em: <www.forbes.com/global2000/list/>. Acesso em 08 maio 2013

FRISCHTAK, C. R. **A Automação Bancária e Mudanças na Produtividade:** A Experiência Brasileira. Pesquisa e Planejamento Econômico, v. 22, nº2, p. 197-240, 1992.

HEGEDUS **Sete novas ferramentas da Qualidade**. 2003. Disponível em: <http://www.geocities.ws/eduardo_turi/303_as_sete_novas_ferramentas.pdf>. Acesso em 17 dez. 2012

KIERAS **Non-Technical Issues in Software Development**. 2009.Disponível em: <http://www.umich.edu/~eecs381/lecture/SoftwareDevelopment.pdf>. Acesso em 17 dez. 2012

MARSHALL JUNIOR, I. *et al.* **Gestão da Qualidade**. Rio de Janeiro. FGV, 2006.

MCCABE, T. J. **A Complexity Measure.** IEEE Transactions on Software Engineering,1976.

MEIRELLES, F. **Empresas brasileiras aplicam 6% do faturamento líquido em TI.** 2009. Disponível em: <http://www.itweb.com.br/noticias/index.asp?cod=57753>. Acesso em 08 maio 2013

MIZUNO, S. **Management for quality improvement – the 7 new QC tools**. Cambridge : Productivity Press, 1988.

MYERS, G. J. **The Art of Software Testing.** 1979.

NEMÉSIO, S. **Cadeia de Valor, Ações Estratégicas e Medição do Desempenho: Uma Abordagem para Organizações de Manutenção.** Dissertação de Mestrado Profissional em Sistemas de Gestão, Universidade Federal Fluminense, Niterói, 2008.

PÁDUA, C. F.; MENDES, M.S **Gestão da Qualidade**. 2009. Disponível em: <www.techoje.com.br/site/techoje/categoria/detalhe_artigo/650>. Acesso em 17 dez. 2012

PORTER, M. **Vantagem Competitiva**. Rio de Janeiro: Ed. Campus, 1995.

SEBRAE **Manual de Ferramentas da Qualidade**. 2005. Disponível em: <http://www.dequi.eel.usp.br/~barcza/FerramentasDaQualidadeSEBRAE.pdf>. Acesso em 17 dez. 2012

ROTONDARO, R. G. **Seis Sigma -Estratégia Gerencial para a Melhoria de Processos, Produtos e Serviços** 1.ed, São Paulo, 2002.

SEACORD, R.; PLAKOSH, D. **Modernizing Legacy Systems: Software Technologies, Engineering Processes, and Business Practices** , Addison Wesley, 2003.

Apêndice A

Levantamento de Não-Conformidades

Prezado Colaborador,

Com o intuito de melhorar a cada dia os serviços prestados pela Gerência Técnica, solicitamos a tua colaboração no preenchimento deste formulário. As informações aqui levantadas serão utilizadas com o intuito de identificar as principais não-conformidades e/ou pontos críticos do nosso processo de fabricação de softwares.
De posse destas informações teremos mais subsídios para definir um plano de ação efetivo para reduzir o retrabalho e aumentar a qualidade de nossos entregáveis.

Não-Conformidade	**Quantidade**	**Gravidade**
Erro de Lógica de Programação		
Requisito de Negócio não atendido		
Requisito de Técnico não atendido		
Indisponibilidade Ambiente de Homologação		
Erro parametrização de Ambiente		

Instruções de preenchimento:
Quantidade: Utilizar valor inteiro que representa a quantidade de cada não conformidade encontrada no período de __/__/__ até __/__/__.E em relação a Não-Conformidade registrar a quantidade unitária "1" para cada período de 30 minutos de indisponibilidade.
Gravidade: Usar valor inteiro de 0 a 10, sendo 0 gravidade mínima e 10 gravidade máxima.

Sugestões

_____

_____

_____

_____

Ponto Focal:	
Nome Software:	

Não Preencher

Preenchimento Área Técnica

☐ Baixa Complexidade ☐ Média Complexidade ☐ Alta Complexidade

Data: / /

Assinatura: _____