

## Ação Corretiva: análise de causa raiz dos defeitos e proposta de um plano de ação

*Cristiane Corrêa da Silveira, Marcelo Carboni Gomes*

**Resumo:** O objetivo deste trabalho é apresentar os resultados referentes ao estudo do método de análise de causa raiz, das técnicas para sua aplicação e a elaboração de um plano de ação para os defeitos de um software financeiro. Para o seu desenvolvimento foi utilizada uma base teórica que apoiou o processo e viabilizou a aplicação deste método.

**Palavras-chave:** Análise de causa raiz; Defeito; Ação corretiva.

**Abstract.** *The objective of this work is to present the results for the study of the method of root cause analysis, techniques for application and elaboration of a plan of action for defects of a financial software. For its development I used a theoretical base that supported the process and made possible the application of this method.*

**Keywords:** *Root cause analysis; Bug; Corrective action.*

### Introdução

Atualmente, a maioria das empresas de TI (Tecnologia da Informação) possui processos de negócio complexos e pouco maleáveis, que ao serem utilizados no decorrer do processo de desenvolvimento de *softwares* permitem o surgimento de anomalias. Segundo PMBOK (2008), processo é um conjunto de ações e atividades inter-relacionadas que são executadas para alcançar um objetivo.

Com a evolução tecnológica das organizações e a crescente exigência por qualidade, muitas pessoas e empresas começam a se preocupar também cada vez mais com a qualidade de seus produtos e serviços, o processo de garantia da qualidade. O CMMI-DEV 1.2 descreve que o propósito da Garantia da Qualidade de Processo e Produto (PPQA) é munir a equipe e a gerência com uma visão clara sobre os processos e seus produtos de trabalho associados.

Para Sommerville (2003), a adequação do processo de desenvolvimento é um dos principais fatores para melhorar a qualidade do produto. Uma vez que os processos são aprimorados para melhorar a qualidade do produto e, em particular, para reduzir o número de defeitos nos softwares fornecidos, a redução dos custos e do tempo pode se tornar a principal meta de melhoria.

O termo defeito muitas vezes é utilizado de forma genérica. No entanto, é importante ter em mente que sua interpretação dependerá do contexto em que ele for utilizado. Estas definições seguem a terminologia padrão para Engenharia de Software do IEEE (IEEE 610.12, 1990):

- Erro: É um defeito cometido por um indivíduo ao tentar entender uma determinada informação, resolver um problema ou utilizar um método ou uma ferramenta;
- Defeito (ou Falta): É uma manifestação concreta de um erro num artefato de *software*. Um erro pode resultar em diversos defeitos;
- Falha: É o comportamento operacional do *software* diferente do esperado pelo usuário. Uma falha pode ter sido causada por diversos defeitos e alguns defeitos podem nunca causar uma falha.

Nesse estudo a nomenclatura utilizada para as anomalias identificadas no *software* será defeito, a divergência no comportamento esperado de um sistema.

Normalmente, quando os defeitos ocorrem, a equipe busca soluções paliativas que permitem a solução momentânea deste em curto prazo. No entanto, ao construir novos projetos, os mesmos problemas voltam a ocorrer devido a ter sido tratado somente o efeito e não a causa do problema (HERAVIZADEH; MENDLING; ROSEMAN, 2008).

Sempre que um defeito volta a ocorrer, o tempo e o custo para sua resolução se tornam maiores (HERAVIZADEH; MENDLING; ROSEMAN, 2008). Isto ocorre, pois a equipe somente realiza a implementação da solução, sem desprender tempo com documentação do processo de resolução do defeito ou analisar, o motivo, frequência ou os principais efeitos da ocorrência deste defeito. No entanto, este quadro pode ser evitado ao ser analisada a causa de origem do problema através do método de Análise da causa raiz.

A Análise de Causa Raiz, também conhecida como RCA (*Root Cause Analysis*) é uma maneira de identificar os motivos que auxiliaram na geração de um problema, afinal os problemas são melhores resolvidos ao tentar corrigir ou eliminar as suas causas. Através deste método, é possível a elaboração e aplicação do plano de ação corretiva e passa-se a monitorar as causas dos defeitos, atuando no processo como um todo antes que estes defeitos voltem a ocorrer (ROONEY & HEWEL, 2004).

Conforme JUCAN (2005), o esforço para identificar a causa raiz de um problema não é um processo fácil, depende da experiência da equipe. Apesar disto, devido ao avanço da tecnologia, este processo tornasse cada vez mais necessário, pois existem cobranças referentes à qualidade do produto entregue, tanto pelo cliente quanto da gerência. Estes solicitam retornos comprovados das causas originais dos defeitos, bem como quais foram às ações utilizadas para resolução e prevenção para que estes defeitos não ocorram novamente.

Já ANDERSEN & FAGERHAUG (2006), definem que o processo de análise de causa raiz permite uma maior agilidade e eficácia para a identificação dos fatos que auxiliaram na ocorrência dos problemas. JUCAN (2005) salienta que o processo de identificação da causa raiz, dependendo da técnica utilizada, pode despende semanas ou meses de avaliação exaustiva. Cada vez mais os projetos de TI são rápidos e precisam de técnicas ágeis para identificação da origem dos seus problemas.

Nas etapas metodológicas deste estudo, se iniciou uma pesquisa bibliográfica sobre a identificação da origem dos seus problemas, o estudo da análise de causa raiz e suas

possíveis técnicas de apoio de aplicação. Diante disto, foi identificado que, para um maior benefício e eficácia, o processo deve ser implantado através de um conjunto de cinco fases: investigação, análise, decisão, comunicação e execução (ROONEY & HEWEL, 2004; JUCAN, 2005).

Com isso, esta pesquisa foi idealizada com o objetivo de realizar um estudo de caso, aplicando a análise de causa raiz em um projeto que apresenta um número elevado de defeitos com criticidade alta, comprometendo assim a qualidade do produto a ser entregue para o cliente. Este estudo propõe verificar a eficácia do método proposto para identificação da causa original, suporte para criação de ações corretivas que visam o crescimento do nível de qualidade dos produtos desenvolvidos. Para tal, apresenta-se uma revisão bibliográfica, pela qual foi realizado um estudo do processo de análise de causa raiz, as principais técnicas para sua aplicação e trabalhos relacionados. Em seguida, é discorrido o processo de análise dos defeitos coletadas na ferramenta de *BugTrack* de um *Software* Financeiro e é apresentado também um plano de ação. Finalmente, avalia-se o plano proposto.

### **Análise de Causa Raiz (RCA)**

A Análise de Causa Raiz, também conhecida como RCA (*Root Cause Analysis*), é um método que permite a identificação e correção dos principais fatores que ocasionaram o problema. Este método visa descobrir os defeitos originais (causa raiz), as quais geraram o problema, ao invés de buscar soluções imediatas para a resolução de um defeito (JUCAN, 2005 ; ANDERSEN & FAGERHAUG, 2006).

Segundo Rooney & Hewel (2004), RCA é uma ferramenta projetada para auxiliar a identificar não apenas “o que” e “como” um evento ocorreu, mas também “por que” ele ocorreu. Somente quando é identificado o motivo original de um defeito ter ocorrido, será viável gerar ações para que este não volte a ocorrer. A utilização da ferramenta RCA segundo Rooney & Hewel (2004), não evita que sempre que ocorrer algum defeito em produção a equipe tenha que buscar soluções imediatas, avaliando somente os sintomas. No entanto, sugere que o defeito seja tratado, mas não seja fechado até que este seja analisado e identificado à causa original que o fez ocorrer.

A Análise de Causa Raiz usa uma terminologia específica (ANDERSEN & FAGERHAUG, 2006), apresentando os seguintes termos para:

- Ocorrência: um evento ou condição que não esteja dentro funcionalidade do sistema normal ou comportamento esperado.
- Evento: Uma ocorrência em tempo real. Fato que pode impactar seriamente no funcionamento do sistema.
- Estado: Qualquer estado do sistema, que pode apresentar implicações negativas para alguma funcionalidade do sistema normal.
- Por que (também chamado de Fator causal): Uma condição ou um evento que resulte ou participe na ocorrência de um efeito. Elas podem ser classificadas como:
  - Causa direta: Uma causa que resultou na ocorrência.
  - Causa contribuinte: A causa que contribuiu para a ocorrência, mas não a fez diretamente.

- Causa raiz: A causa que, se corrigida, impedirá o retorno desta e de ocorrências similares.
- Cadeia de fatores causais (sequência de eventos e fatores causais): Uma sequência de causa e efeito em que uma ação específica cria uma condição que contribui ou resulta em um evento. Isso cria novas condições que, por sua vez, resultam em outros eventos.

### Técnicas

Para a aplicação do RCA no estudo de caso, foi utilizada a combinação de técnicas, permitindo uma maior exatidão na identificação da causa raiz, conforme abaixo descritas:

- Diagrama de Causa e Efeito, também conhecido como diagrama de *Ishikawa* (espinha de peixe): permite identificar, explorar e apresentar graficamente todas as possíveis causas relacionadas a um único problema. Esta técnica é utilizada em equipe e permite classificar os defeitos em seis tipos diferentes de categorias: método, matéria-prima, mão-de-obra, máquinas, medição e meio ambiente. Sendo, que nem o número e nem os tipos de categorias são pré-estabelecidas, permitindo a equipe adequar estes conforme sua necessidade. E, através desta técnica, se torna possível à identificação das causas potenciais de determinada defeito ou oportunidade de melhoria, bem como seus efeitos sobre a qualidade dos produtos. Além disto, ela permite também estruturar qualquer sistema que necessite de resposta de forma gráfica e sintética (isto é, com melhor visualização) (JUCAN, 2005).
- Cinco Porquês: baseada na realização de cinco iterações de perguntas às quais, é colocado em questão o porquê daquele problema, sempre questionando a causa anterior. O número de cinco perguntas é variável, pois na prática pode ser identificada a causa raiz do problema através de mais de cinco perguntas ou menos de cinco perguntas (SERRAT, 2009).
- Reunião de Análise Causal: as causas dos problemas são levantadas em reuniões do tipo “*Brainstorming*”. As causas mais prováveis são discutidas entre a equipe, e posteriormente a descoberta das prováveis causas, os participantes propõem ações corretivas para estes problemas no futuro (JUCAN, 2005).

### Trabalhos relacionados

A análise de causa raiz é utilizada em diversas áreas, não sendo um método exclusivo da área de tecnologia da informação, esta pode ser aplicada na área médica (TEXEIRA, 2007), na de sistemas elétricos (PIRES, 2010), na industrial (ROONEY & HEWEL, 2004), entre outras, com o propósito de interpretar o conjunto de eventos observados e apontar a origem que está causando estes eventos, i.e., onde tudo começou.

Em “Análise de Causa Raiz em Processos de Negócio” (HERAVIZADEH, MENDLING, TOSEMANN, 2008), foi aplicada a técnica de análise de causa raiz baseadas em modelos de processos de negócio. Este trabalho apresenta a utilização da análise de causa raiz como uma metodologia sistemática para detectar e documentar a dimensão qualitativa de um processo de negócio. Neste, foi criado uma base para a análise de causa raiz em modelagem de processos de negócio e uma integração conceitual entre abordagens baseadas em metas

e abordagens baseadas em atividades para entender processos. Além disto, algumas normas e modelos de maturidade (por exemplo, ISO/IEC 12207, CMMI-DEV e MR-MPS) também recomendam a execução da análise de causas, confirmando a importância deste processo em uma organização de alta maturidade, ou seja, que possui seus processos bem definidos e busca a melhoria contínua.

A norma ISO/IEC 12207 (ISO/IEC, 2008) apresenta indiretamente o termo "análise de causas", dentro do processo "Resolução de Problemas de *Software*". Este processo tem o propósito de garantir que todos os problemas sejam identificados, analisados, gerenciados e controlados até sua resolução. Na ISO/IEC 15504-5 (ISO/IEC, 2003), a execução da análise de causas é apresentada como uma prática base do resultado "Problemas são analisados e avaliados para identificar soluções aceitáveis" do processo "Gerência de Resolução de Problemas". Assim como na norma ISO/IEC 15504, o CMMI-DEV (SEI, 2006), no nível "4" e "5", e o MRMPMS (SOFTEX, 2009<sup>a</sup>), no nível "A" e "B", também recomendam a utilização do método de análise de causa raiz.

### **Metodologia e aplicação do RCA**

Como metodologia para este estudo, foi realizada uma pesquisa aplicada de forma qualitativa através de um estudo de caso para um *software* financeiro de uma empresa situada na cidade de Porto Alegre, estado do Rio Grande do Sul, cujo ramo é alimentício e varejista.

O *software* financeiro, em questão possui funcionalidades destinadas para o gerenciamento e controle financeiro das vendas realizadas pela empresa (entrada e saída de operadoras, controle de caixa, entre outras). Por questões de normas internas de confidencialidade da empresa em estudo, tanto sua identidade quanto quaisquer informações técnicas estratégicas de seus produtos estão sendo preservados. Foi identificado que para um maior benefício e eficácia, o processo foi implantado através de um conjunto de cinco fases: investigação, análise, decisão, comunicação e execução (ROONEY & HEWEL, 2004; JUCAN, 2005).

### **Investigação**

Esta fase tem o objetivo de identificar de forma neutra os fatos que apresentam como o defeito aconteceu. Para isto, devem ser coletados os principais dados referentes à ocorrência do defeito (Ex.: como o defeito ocorreu, quais as ações realizadas para sua correção imediata, passos para sua identificação).

A aplicação foi dividida em duas etapas: levantamento total dos defeitos e a seleção somente de uma amostragem para aplicação nesta proposta. Esta divisão foi necessária, devido ao *software* apresentar um elevado número de defeitos e também pelo tempo limitado na entrega desta proposta.

Na primeira etapa, foi realizado o levantamento referente ao número de ocorrências do projeto em questão, através da ferramenta de *BugTrack* utilizado no *software* financeiro, apresentando os seguintes números identificados através da criticidade (alta, média, baixa):

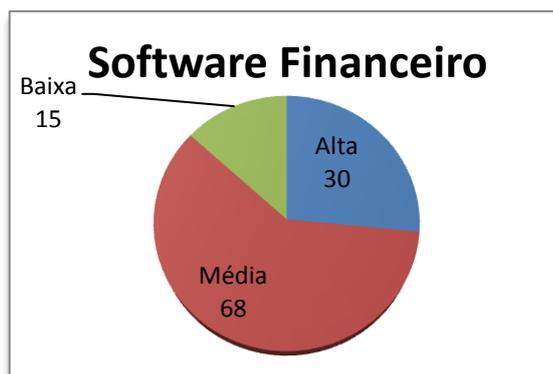


Figura 1 - Etapa 1: Levantamento de defeitos do *Software Financeiro*

Pelo número de defeitos apresentadas no desenvolvimento do *software* financeiro e objetivando uma melhor efetividade do trabalho, foi aplicada a segunda etapa da investigação. Onde foram selecionados somente os defeitos que apresentaram criticidade alta, agrupando os mesmos por similaridade (aqui será utilizada a similaridade para os defeitos com as mesmas características). Segue, na Tabela 1, um exemplo do detalhamento de defeitos agrupados por similaridade.

Tabela 1 — Etapa 2: Exemplo da Aplicação do Detalhamento dos defeito

Defeitos	Descrição (como ocorreu)	Passos para Reprodução	Ações realizadas
Erro ao validar os campos "Banco" e "Agência"	Após um consulta por um cheque, foi alterado o código do banco. Posteriormente, foi inserido no campo "agência", o número de uma agência que exista no software cadastrado para outro banco. Sistema apresenta a mensagem de erro: "javax.persistence.NonUniqueResultException: br.com.zaffari.tsr.server.entity.AgenciaBancaria"	<ol style="list-style-type: none"> <li>1) Logar usuário em local conectado a empresa padrão do Software Financeiro.</li> <li>2) No menu "Movimento", clicar na opção "Controle de Cheques".</li> <li>3) Informar nos campos "Banco", "Agência", "Conta CMC7", "Nº do cheque" e "Local recebimento" valores correspondentes a um cheque cadastrado no sistema e disparar a ação de [Consultar].</li> <li>4) Informar um código de banco não cadastrado no sistema no campo "Banco"</li> <li>5) No campo "Agência", informar o código 25 e voltar o foco para o campo "Banco".</li> </ol>	<ol style="list-style-type: none"> <li>1) Desenvolvedor simulou o erro</li> <li>2) Desenvolvedor identificou o erro na implementação referencia a regra de dependência entre os campos banco e agência</li> </ol>
Erro ao incluir agência não cadastrada no sistema	Ao acessar a tela de Controle de Cheques e informar uma agência não cadastrada no sistema, o mesmo não esta apresentando a informação de agência não cadastrada	<ol style="list-style-type: none"> <li>1) Logar com usuário em local conectado a empresa padrão do Software Financeiro</li> <li>2) No menu "Movimento", clicar na opção "Controle de Cheques"</li> <li>3) Informar uma agência não cadastrada no sistema</li> </ol>	<ol style="list-style-type: none"> <li>1) Desenvolvedor simulou o erro</li> <li>2) Desenvolvedor identificou que o erro ocorria devia ter erro na validação do campo agência.</li> </ol>

Na Tabela 1, observa-se o agrupamento e a similaridade entre os defeitos. A similaridade na validação do campo agência foi identificada em mais de um processo de validação do *software* financeiro.

## Análise

Esta fase tem o objetivo de apresentar as razões que demonstram por que ocorreu o defeito. Para a aplicação desta fase, foi utilizada a combinação das técnicas "Diagrama de Causa e Efeito", "Cinco porquês" e reuniões de "*Brainstorming*". Para a realização desta fase foram agendadas quatro reuniões de "*Brainstorming*" com integrantes da equipe que participou do processo de desenvolvimento do *Software* Financeiro (desenvolvedor, analista de testes e o gestor).

Na primeira reunião, foi apresentada ao grupo a proposta deste estudo, assim como a tabela de defeitos gerados na fase de investigação. Posteriormente, foi desenhada, utilizando o recurso de um quadro, a "espinha de peixe", referente ao diagrama de *Ishikawa* (JUCAN, 2005). Com o auxílio dos integrantes, foram denominadas as categorias principais (comunicação, infraestrutura, processo, ambiente/organização, recursos humanos e gestão), ver Figura 2.

Após a denominação das categorias, se iniciou o processo de análise do primeiro defeito ("efeito"), incluindo o mesmo no diagrama de *Ishikawa*. Com o auxílio dos integrantes, foi realizado o processo de análise dos motivos que ocasionaram o defeito, onde cada integrante informou as causas potenciais que auxiliaram para ocorrência deste defeito. Ao final da identificação das causas, com o auxílio da ferramenta "Cinco porquês" (SERRAT, 2009), foi realizado o questionamento do "por que" aquela causa ocorreu e documentado na espinha de peixe (ver Figura 2), como uma sub-causa. Este processo somente foi finalizado, quando se obtinha resposta para todos os "porquês", e quando todas as causas atingiam uma causa raiz.

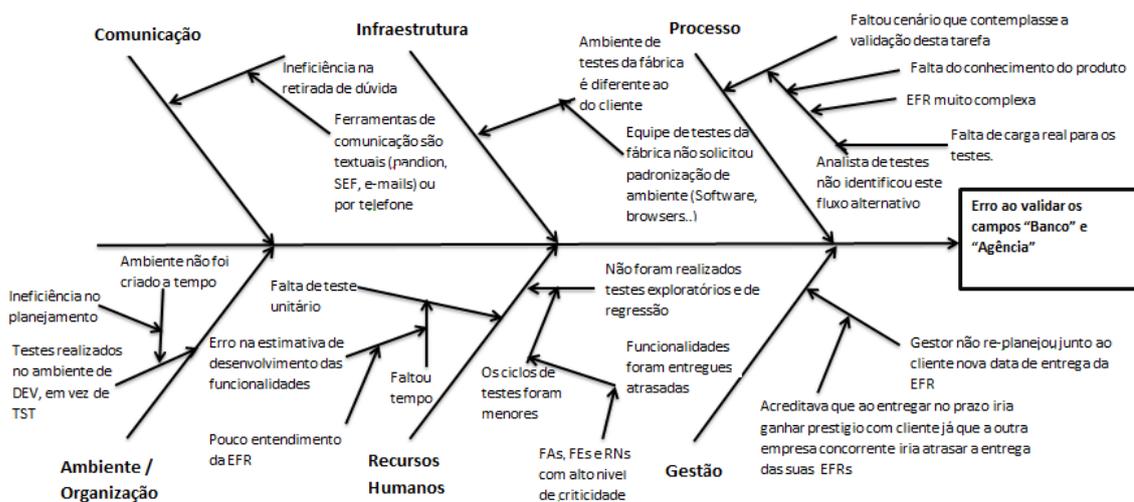


Figura 2 - Exemplo do Diagrama de Ishikawa

Esta técnica foi aplicada nas demais reuniões de "*Brainstorming*", realizando assim a análise de todos os defeitos com criticidade alta.

## Decisão

Esta fase tem o objetivo de identificar as ações e as lições aprendidas para corrigir ou eliminar as causas de um defeito, a fim de alcançar em longo prazo, resultados efetivos. Nesta fase, foi criada uma planilha agrupando por similaridade todas as causas raízes identificadas na fase de Análise. Além disto, foi realizada uma reunião de "*Brainstorming*" com os integrantes (mesmos utilizados na fase anterior), onde foi apresentada a relação de causas raízes identificadas. Para complementar este estudo, foram solicitadas sugestões de melhorias associadas a cada item, através da aplicação da técnica de "Reunião de Análise Causal" (JUCAN, 2005).

**Tabela 2 — Exemplo: Aplicação da Técnica de “Reunião de Análise Causal”**

Problema: Erro ao validar os campos “Banco” e “Agência”	
Causas	Ações de Melhorias Sugeridas
Ferramentas de comunicação são textuais	Realizar reuniões semanais e presencias para retirada de dúvidas das EFRs <sup>1</sup> e caso seja identificado a necessidade de alteração da EFRs, representante da fábrica cria uma SEF <sup>2</sup> .
Equipe de testes da fábrica não solicitou padronização de ambiente ( <i>Softwares, browsers...</i> )	No início dos projetos, realizar um levantamento dos programas instalados nas máquinas de homologação e criar ambiente parecido ao utilizado (sistema operacional, <i>browsers...</i> ).
EFR muito complexa FAS <sup>3</sup> , FES <sup>4</sup> e RNS <sup>5</sup> com alto nível de criticidade Pouco entendimento da EFR Falta do conhecimento do produto	Envolver o analista de testes em todas as reuniões (retiradas de dúvidas, reunião de apresentação das funcionalidades da EFR...) Incluir no processo a atividade de revisão de CTs <sup>6</sup> por outro integrante do projeto. Realizar reuniões semanais e presencias para retirada de dúvidas das EFRs e caso seja identificado à necessidade de alteração da EFRs, representante criava uma SEF.
Falta de carga real para os testes.	Para uma validação/verificação das funcionalidades se faz necessário dados reais. Diante disto, inserir no processo, a solicitação de dados reais para a base de testes.
Ineficiência no planejamento	Incluir no plano de projeto/vendas, pré-requisitos com data de prazos, normas... Caso não seja atendido o prazo, apresentar os impactos que os mesmos podem causar e tentar encontrar junto ao gerente ações a serem tomadas.
Acreditava que ao entregar no prazo iria ganhar prestígio com cliente já que a outra empresa concorrente iria atrasar a entrega das suas EFRs	Apresentar relatórios periódicos sobre o status de testes, mostrando o valor da qualidade de um software na entrega.

<sup>1</sup> EFR – Sigla denominada para a nomenclatura “Especificação Funcional do Requisito”

<sup>2</sup> SEF – Sigla denominada para a nomenclatura “Solicitação de Esclarecimento ao Fornecedor”

<sup>3</sup> FA – Sigla denominada para a nomenclatura de “Fluxo alternativo” (Especificação Funcional)

<sup>4</sup> FE – Sigla denominada para a nomenclatura de “Fluxo de Exceção” (Especificação Funcional)

<sup>5</sup> RN – Sigla denominada para a nomenclatura de “Regra de Negócio” (Especificação Funcional)

<sup>6</sup> CT – Sigla denominada para a nomenclatura “Caso de Teste”

## Comunicação

Esta fase tem o objetivo de comunicar aos integrantes envolvidos no processo de análise dos defeitos, os resultados obtidos, assim como as ações corretivas propostas. Nesta fase foi agendada uma reunião com todos os integrantes da equipe do *Software* Financeiro. Na reunião foi apresentado todo o processo RCA executado e o plano de ação criado, utilizando as ações de melhorias sugeridas na fase de Decisão. Além disto, foi discutido com a equipe a eficácia das ações e sua aplicação.

**Tabela 3 — Plano de Ação (Ações Corretivas)**

Ações	Responsáveis
1) No início dos projetos, criar documento apresentando o processo a ser utilizado, inclusive o de testes, apresentando responsabilidades, riscos (com associação de prazos). Deixar claro, os riscos associadas ao não cumprimento das responsabilidades e prazos pré-estabelecidos (esta permitira um maior conforto da equipe da fábrica para realizar o processo de cobrança de pendências e também justificará atrasos de entrega ou nível menor da qualidade do produto entregue).	<u>Principal:</u> Gerente <u>Secundários:</u> Analista de Testes e Desenvolvedor
2) Um requisito/funcionalidade somente poderá ser entregue, posteriormente a liberação do responsável da equipe de testes. Caso o gerente queira liberar este sem esta liberação, POR EXIGÊNCIA DO CLIENTE, deverá ser realizada uma reunião com o cliente final, apresentando os erros conhecidos, percentual de cobertura de testes, riscos associados a entrega. Nesta reunião, deverão ser apresentados estes dados ao cliente, questionando se ele aceita o produto com o quadro apresentado. Caso o cliente aceite, deverá ser documentado este aceite e serão de responsabilidade do dele os riscos associados ao quadro apresentado. Esta documentação pode ser realizada através de um e-mail ou documento formal e deve ser encaminhada ao gerente do projeto, gerente de testes e cliente final.	<u>Principal:</u> Gerente e Analista de Testes <u>Secundários:</u> Cliente final, Gerente de testes
3) Um requisito/funcionalidade somente poderá ser entregue, posteriormente a liberação do responsável da equipe de testes. Caso o gerente queira liberar este sem esta liberação, SEM EXIGÊNCIA DO CLIENTE, deverá ser documentado que a equipe de testes não aconselha a liberação ao cliente, apresentando os motivos, erros conhecidos, percentual de cobertura de testes, riscos associados à entrega. Esta documentação pode ser realizada através de um e-mail ou documento formal e deve ser encaminhada ao gerente do projeto, gerente de testes e representante superior responsável pelo cliente em questão na empresa.	<u>Principal:</u> Gerente e Analista de Testes <u>Secundários:</u> Gerente de testes
4) Sempre realizar a solicitação de atividades ou aviso de itens pendentes com um prazo de no mínimo uma semana de antecedência.	<u>Principal:</u> Integrante da equipe que esta realizando a solicitação
5) Sempre que for identificado que não será possível cumprir com os prazos estabelecidos por deficiências de recursos, conhecimento..., deve ser alinhado ações alternativas, visando minimizar os impactos no projeto.	<u>Principal:</u> Gerente
6) No início do projeto, realizar as seguintes ações: - Encaminhar as EFRs para um integrante da equipe de	<u>Principal:</u> Desenvolvedor e Analista de Testes

desenvolvimento e um integrante da equipe de testes (recursos já planejados para projeto em questão) - Integrantes devem realizar o processo de leitura, entendimento e validação da EFR, listando dúvidas encontradas (aplicar checklist padrão de validação de EFRs para auxiliar no trabalho) - Realizar reunião de no máximo 1 hora entre os 2 integrantes para discutir duvidas, regras, FE's, FA's... - Marcar reunião com o analista de sistemas para solucionar as duvidas identificadas (2 integrantes participam da reunião)	<u>Secundário</u> : Analista de Sistemas
7) Criar checklist padrão de validação de EFRs e CTs	<u>Principal</u> : Analista de testes e Desenvolvedor
8) Realizar reunião semanal para retirada de dúvidas referente EFRs. Tempo = N° EFRs que possuem duvidas * tempo médio estipulado (Sugerido = 0,25 / 15 min) Reunião deve ocorrer no primeiro dia útil da semana. Caso não haja dúvidas referentes à semana que passou, não ocorre reunião. Modificações identificadas devido às dúvidas geraram abertura de SEF.	<u>Principal</u> : Analista de Testes, Desenvolvedor e Analista de Sistemas <u>Secundário</u> : Gerente
9) Todas as reuniões referente a EFR, sempre deverá ir um integrante da equipe de desenvolvimento e um integrante da equipe de testes	<u>Principal</u> : Desenvolvedor e Analista de Testes
10) Planejar a alocação dos recursos do projeto sempre prevendo um líder técnico sênior (tecnologia e conhecimento do produto) e um analista de testes sênior (conhecimento de testes e produto). Em ambas as áreas, a pré-condição de nível sênior em conhecimento do produto e do método não precisa estar em uma única pessoa.	<u>Principal</u> : Gerente <u>Secundários</u> : Analista de testes e Desenvolvedor
11) No inicio do projeto, validar a possibilidade de ser realizado uma atualização da base de dados dos ambientes de desenvolvimento e testes com dados reais (produção) permitindo um nível maior de precisão na validação das regras e fluxos	<u>Principal</u> : Gerente
12) Caso ação do item 11 não seja possível: - No inicio do projeto, validar com o cliente se carga de dados será realiza por ele. Caso afirmativo, incluir datas de prazo para a entrega das necessidades referente a dados e, data de prazo para a carga estar criada.	<u>Principal</u> : Gerente
13) Caso a ação do item 11 não seja possível, ação do item 12 não possa ser realizada e na ação do item 5, o cliente realize a compra dos serviços de geração de carga: Criar carga consistente, levando em consideração todo o processo do produto (ex.: outras EFRs que utilizam os mesmos dados, preenchimento de todos os campos, nomenclaturas adequadas...) simulando dados reais.	<u>Principal</u> : Gerente e Desenvolvedor
14) Na construção dos cenários de testes, sempre que identificado a necessidade de validação dos dados via banco, construir e incluir as queries para validação	<u>Principal</u> : Analista de Testes
15) Diariamente, enviar relatório com andamento dos testes para a equipe do projeto e superiores No relatório deverão ser apresentadas informações referentes EFRs do escopo de testes (N° Total de CTs planejados X N° Total de CTs	<u>Principal</u> : Analista de Testes

testados, N° de CTs que passaram, falharam, estão bloqueados, relação de erros/melhorias encontradas, considerações...)	
16) Antes do início da fase de testes, realizar o processo de revisão dos CTs das EFRs com criticidade > = Alta. O processo de revisão dos CTs deve ser realizado por outro integrante da equipe de projeto e que possua perfil de análise.	<u>Principal</u> : Revisor (Analista de testes, líder técnico, analista de sistemas...)
17) No início do projeto, encaminhar solicitação a equipe de infraestrutura para geração de ambiente de testes. O ambiente de testes deverá possuir as mesmas configurações de software do ambiente de homologação (Ex.: Sistema operacional, browsers (versão e tipos)...)	<u>Principal</u> : Analista de testes <u>Secundário</u> : Gerente
18) Evitar realizar rotatividade de recursos ou possuir recursos com alocação <i>Part-Time</i>	<u>Principal</u> : Gerente
19) Gerente/ gestor deverá realizar um trabalho de acompanhamento da equipe, tentando identificar sintomas referente a desmotivação, desatenção, ... Sempre que for identificado estes sintomas, conversar com o recurso. Caso identificar que sintoma não mudou ou que voltou a ocorrer, encaminhar para o Recurso Humano realizar um trabalho motivacional com o recurso.	<u>Principal</u> : Gerente <u>Secundário</u> : Recurso Humano
20) O código somente deverá ser liberado posteriormente a aplicação do <i>checklist</i> de desenvolvimento e atingir o percentual pré-estabelecido de teste unitário sobre linha de código	<u>Principal</u> : Desenvolvedor
21) Criar base histórica com tempos de estimativas. Realizar a manutenção desta base no término de cada projeto, permitindo aumento de assertividade no tempo das estimativas de desenvolvimento, testes...	<u>Principal</u> : Desenvolvedor e Analista de Testes

## Execução

Esta fase tem o objetivo de implementar as ações identificadas durante a fase de decisão, acompanhar as atividades e identificar a eficácia das ações corretivas. Para esta fase estava previsto a aplicação das ações corretivas apresentadas na fase de comunicação em uma nova etapa do *Software* Financeiro. No entanto, decorrente de modificações internas na estrutura (implantação de um novo repositórios de dados), todos os desenvolvimentos de novas funcionalidades do software foram colocados em "*Standby*". Diante disto, foi realizado um replanejamento do método de avaliação, onde foi gerado um documento com questões a serem aplicadas a um grupo específico, a fim de avaliar a viabilidade e eficácia das ações planejadas e do método de análise de causa raiz.

O andamento desta fase ocorreu através das seguintes etapas:

- Escolha dos integrantes do grupo de avaliação das ações corretivas: Para a escolha dos avaliadores foi analisado quais os integrantes possuíam maior senso crítico, familiaridade com o processo de desenvolvimento do software e um nível maior de impactos com as ações planejadas. Diante disto, foram escolhidos os integrantes que realizavam ações

gerenciais ou de liderança, tais como: um Analista de Sistema, um Analista de Testes, um Gerente de Testes, um Gerente do Projeto e um Líder Técnico.

- Execução do processo: Durante a reunião de comunicação (fase anterior) foi explicada à necessidade da avaliação das ações propostas, assim como do processo utilizado, e informado que seriam enviados via e-mail a alguns integrantes da equipe um questionário para avaliação das ações apresentadas. Cada avaliador respondeu a perguntas fechadas, essas perguntas são de autoria própria com base em SAMARA(2007).
- Avaliação do questionário: as respostas das perguntas foram agrupadas e analisadas, avaliando o retorno do estudo. Os resultados referentes a esta análise serão apresentados a seguir.

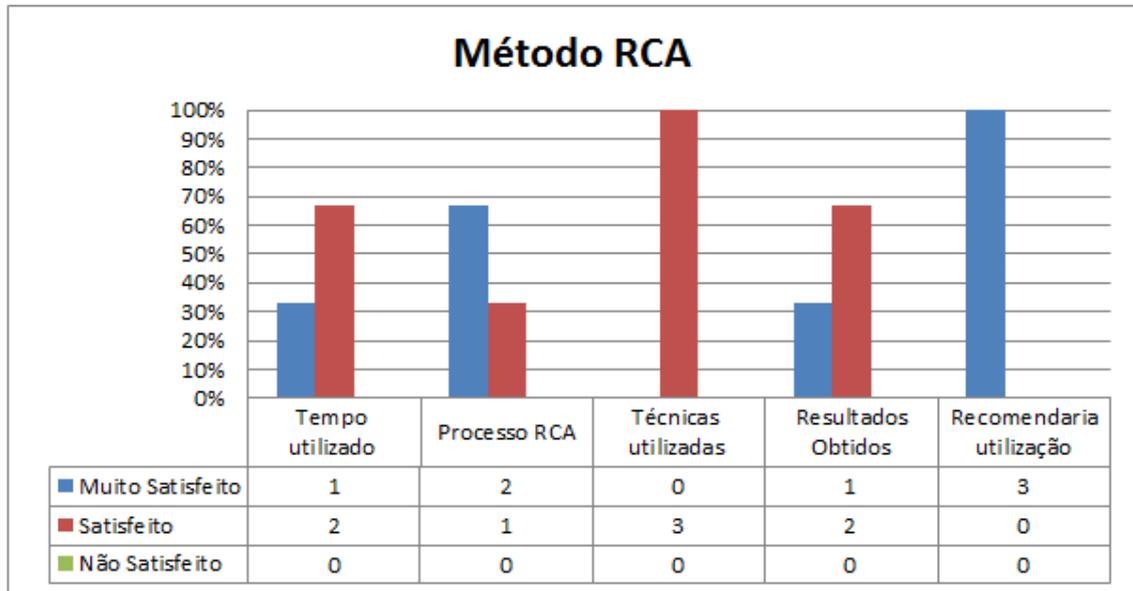
### **Avaliação do Método RCA e a Ação Corretiva**

Visando avaliar o método RCA para o estudo em questão e também as ações corretivas propostas, foram aplicados dois questionários específicos a grupos com diferentes integrantes. Os questionários serão descritos a seguir.

O primeiro questionário, que se referia a questões relativas ao nível de satisfação do método RCA aplicado, foi encaminhado para os integrantes da equipe que participou do processo. Através deste, os integrantes escolheram as alternativas que melhor demonstravam o seu nível de satisfação ("Muito Satisfeito", "Satisfeito" ou "Insatisfeito"):

- Você considera que o tempo utilizado na execução do processo de análise da causa raiz foi satisfatório?
- Em sua opinião, a utilização do processo de análise de causa raiz auxiliou na identificação dos motivos originais dos erros?
- Em sua opinião, as técnicas utilizadas (diagrama de *Ishikawa*, Cinco Porquês e reunião *Brainstorming*) facilitaram na visualização e identificação das causas das ocorrências dos problemas?
- De acordo com sua opinião, os resultados (ações corretivas) obtidos pela análise de causas foram satisfatórios?
- Você recomendaria a utilização deste método a outras equipes da empresa? (Sendo, "Muito Satisfeito = Sim" / "Satisfeito = Talvez" / "Insatisfeito = Não").

A pesquisa apresentou o seguinte resultado:

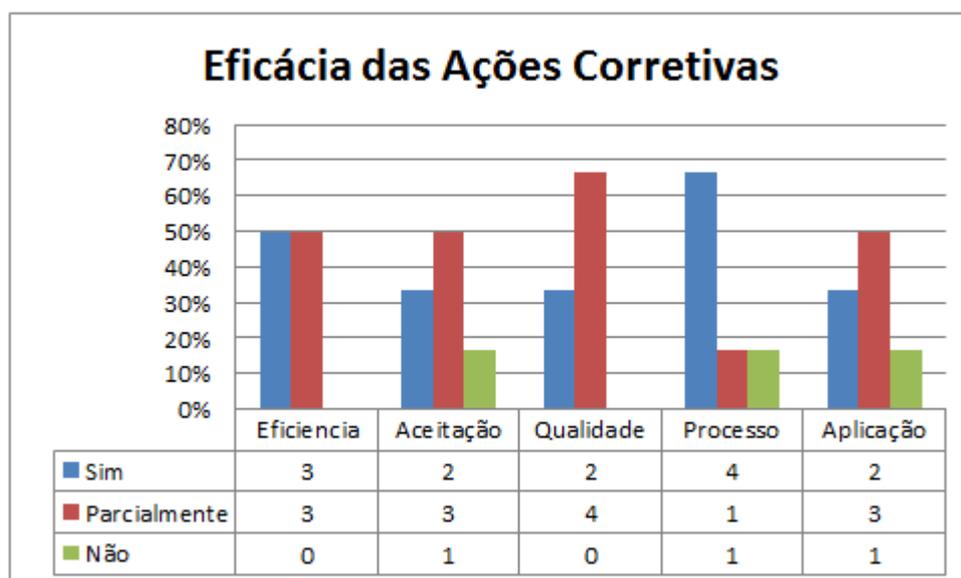


**Figura 3 - Resultado da Pesquisa de Satisfação do Método RCA**

O segundo questionário, que possuía questões referentes ao nível de eficácia das ações corretivas propostas, foi encaminhado à equipe de avaliação (definida na fase de execução do processo RCA). Através deste questionário, os avaliadores determinaram o nível de confiabilidade referente à efetividade das ações corretivas propostas para a melhoria do processo de desenvolvimento do *Software* Financeiro. O questionário apresentava as seguintes perguntas:

- As ações proposta são eficientes para minimizar ou sanar a recorrência de defeitos?
- As ações propostas terão uma boa aceitação no nível de equipe?
- As ações propostas aumentaram o nível de qualidade do produto entregue?
- As ações propostas auxiliaram a equipe, permitindo um melhor andamento das atividades de desenvolvimento, testes e gerência?
- As ações propostas são fáceis de serem aplicadas?
- Teria alguma ação que não foi identificada, mas que você sugeriria, visando melhoria do processo de desenvolvimento de sistemas?

Dessa forma:



**Figura 4 - Resultado da Pesquisa de Eficácia das Ações Corretivas**

Quanto à pergunta “Teria alguma ação que não foi identificada, mas que você sugeria, visando melhoria do processo de desenvolvimento de sistemas?”, apresentou as seguintes ações:

- Incluir uma ferramenta para inclusão de lições aprendidas, para que os integrantes da equipe a preencham durante o processo de execução.
- Realizar o processo de revisão da documentação de testes, utilizando como revisor um integrante da equipe.
- Incluir na ferramenta de BugTrack um campo referente a causa raiz. Este campo será utilizado para a inclusão da razão original da ocorrência no término do processo RCA. Conforme sugestão, esta informação permitirá uma maior facilidade na geração de relatórios e uma base histórica para consultas futuras.

### Considerações Finais

Como principal contribuição desta pesquisa, a realização do estudo utilizando um método direcionado para a identificação dos motivos causadores pela ocorrência dos defeitos, um das principais preocupações quando se referencia a qualidade do produto. Além disto, este estudo permitiu uma maior conscientização para a equipe participante quanto à qualidade do produto desenvolvido. Isto ocorreu, pois ao realizar a análise dos defeitos, os integrantes acabavam sutilmente realizando uma autoavaliação do seu desempenho, permitindo assim aprendizado e motivação para melhorar suas ações. No entanto, para obter sucesso na realização deste é imprescindível que o mesmo seja realizado com a equipe envolvida no software em desenvolvimento. Isto ocorre, pois cada indivíduo possui conhecimentos e críticas próprias referentes ao processo de resolução de problemas. Esta diferenciação entre os indivíduos permite que surjam soluções e pontos críticos antes não identificados. Além disto, é importante que a equipe de análise da causa raiz possua sempre integrantes com funções diferenciadas (analista de sistemas, analista de testes, desenvolvedores, líder

técnico... ), evitando que as ações sugeridas somente solucionem ou que os problemas acabem sendo direcionados somente para uma determinada área.

Saliente-se que a eficácia de resolução de defeitos através das técnicas de análise de causa raiz representa, para a maioria das organizações, uma mudança significativa na sua cultura. E, que provavelmente, irão surgir muitos integrantes da equipe que irão inserir barreiras para não executar estas ações corretivas identificadas. E, é evidente que os ganhos referentes à execução deste método, dependem diretamente do empenho e comprometimento da equipe para encontrar as causas originais dos problemas, e também da disciplina desta para cumprir as ações corretivas identificadas.

Em relação a perspectivas futuras, entende-se que as ações corretivas identificadas devem ser aplicadas em uma nova fase de desenvolvimento do *Software* Financeiro, permitindo uma avaliação mais efetiva. Outro ponto a se destacar é a utilização de uma ferramenta para inclusão das ações corretivas em um local que esteja disponível para acesso comum (Ex.: WIKI, Sharepoint, etc.).

É interessante como trabalhos futuros utilizar outros métodos de análise dos defeitos e aplicando em diferentes projetos de desenvolvimento de software, avaliando a efetividade dos mesmos em relação à análise de causa raiz, assim como para identificação de ações que permitam soluções diferenciadas.

### Referências

ANDERSEN, B., FAGERHAUG, T. (2006), Root Cause Analysis: Simplified Tools and Techniques. ASQ Quality Press

ECKERT, Chris(2005), Apollo Análise de Causa de Raiz (RCA). SP

HERAVIZADEH, Mitra; MENDLING, Jan; ROSEMAN, Michael (2008), Análise de Causa Raiz em Processos de Negócio. SP

IEEE(1991), IEEE standard glossary of software engineering terminology, IEEE Std 610.12-1990, New York

ISO/IEC (2003), ISO/IEC 15504: Information Technology – Software Process Assessment, Parts 1-5, The International Organization for the Standardization and the International Electrotechnical Commission.

ISO/IEC (2008), ISO/IEC 12207: System and Software Engineering – Software Life Cycle Processes, The International Organization for the Standardization and the International Electrotechnical Commission.

JUCAN, George (2005), Root Cause Analysis for IT Incidents Investigation. Toronto, Ontario.

PIRES, Stéfani Pires (2010), Descoberta de Causa-raiz em ocorrências de Sistemas Elétricos. Campinas Grande, Paraíba

PMBOK (2008), 4ª Edição, PMI Book Service Center, Pensylvania

PRODANOV, C. (2004), Manual de metodologia científica. FEEVALE

ROONEY, J.J. & HEWEL, L.N.V.(2004), Root cause analysis for beginners. Quality Progress.

SAMARA, Beatriz dos Santos. BARROS, José Carlos (2007), Pesquisa de Marketing: Conceitos e Metodologia. São Paulo: Prentice Hall

SEI (2006), CMMI® for Development (CMMI-DEV), V1.2, CMU/SEI-2006-TR-008, Software Engineering Institute. Disponível em: <http://www.sei.cmu.edu/>. Acesso em: Julho/2011.

SERRAT, Oliver (2009) The five whys technique. Knowledge Solutions

SOMMERVILLE, Ian (2003) Engenharia de Software. 6. ed. São Paulo: Addison Wesley

SOFTEX (2009a), “MPS.BR – Melhoria de Processo do Software Brasileiro – Guia Geral”. Disponível em: <http://www.softex.br/mpsbr>. Acesso em: julho/2011.

TEXEIRA, Thalyta Cardoso Alux (2007), Análise de causa raiz dos erros de medicação em uma unidade de internação de um hospital universitário. Ribeirão Preto. SP