

Sistema de Análise de Incidentes para Melhoria Contínua

Lucas Filippi
Unisinos
São Leopoldo, Brasil
lucas.filippi2@gmail.com

Sandro José Rigo
Unisinos
São Leopoldo, Brasil
rigo@unisinos.br

Abstract — Software quality is essential for software companies maintain their competitiveness. To achieve continuous improvement in quality, companies can rely on various techniques and tools. The proper use of these tools can provide the company with many benefits such as reduce costs, increase productivity, reduce failures, increase product reliability and customer satisfaction. This paper presents some of these techniques or tools and suggests the use of them together with an application to help a team to reduce the time spent on support and maintenance. The improvement proposal will be elaborated through the PDCA cycle resulting in a system that uses quality tools with manual and automated steps. The use of the system helps the team to identify and prioritize improvement and preventive action opportunities through information taken from the incidents reported by customers, fostering continuous improvement.

Keywords—*Information systems; Quality Tools; PDCA; Decision Support; Continuous improvement*

Resumo — Qualidade de software é indispensável para as empresas do setor se manterem competitivas. Para alcançar uma melhoria contínua na qualidade, as empresas podem contar com vários tipos de técnicas e ferramentas. A utilização adequada destas ferramentas pode proporcionar à empresa diversos benefícios, tais como redução de custos, aumento da produtividade, redução de falhas, melhora da confiabilidade do produto e satisfação dos clientes. Este trabalho apresenta algumas dessas técnicas ou ferramentas e sugere a utilização destas em conjunto com uma aplicação para auxiliar uma equipe a reduzir o tempo gasto com suporte e manutenção. A proposta de melhoria será elaborada através do ciclo PDCA resultando em um sistema que utiliza ferramentas de qualidade, com passos manuais e automatizados. A utilização do sistema auxilia a equipe a identificar e priorizar oportunidades de melhoria e ações preventivas através da informação retirada dos incidentes reportados pelos clientes, promovendo assim o melhoramento contínuo.

Palavras-chave—*Sistema de Informação; Ferramentas de Qualidade; PDCA; Apoio à Decisão; Melhoria Contínua*

I. INTRODUÇÃO

O mercado atual está cada vez mais competitivo, os consumidores mais exigentes, existe grande variedade de produtos disponíveis e novas tecnologias entram no mercado

rapidamente. Por isso as empresas buscam melhores resultados reduzindo custos e aumentando sua qualidade e produtividade.

Conforme o mercado de software cresce e a demanda de novos desenvolvimentos vai aumentando, o desafio é atender essa demanda sem sacrificar a qualidade [8]. Muitas empresas de T.I. gastam um tempo considerável com suporte e manutenção, tempo esse que poderia ser gasto em qualidade, novos desenvolvimentos e inovação. Além disso, alto número de incidentes é sinônimo de clientes insatisfeitos. Independente da definição de qualidade que a empresa utiliza, a maioria dos profissionais concorda que a presença de defeitos indica falta de qualidade [7].

A empresa objeto deste estudo é uma grande empresa do ramo de tecnologia da informação. Foi identificada em uma das equipes dessa empresa, a necessidade de reduzir o esforço de manutenção e o número de incidentes, tratados como mensagens textuais. Essa equipe é responsável por uma pequena parte do principal produto, atuando na área de localização. Recentemente, a equipe passou por um aumento no seu escopo, entretanto, não houveram novas contratações, resultando em um maior esforço da equipe para suprir as necessidades de novos desenvolvimentos e manutenção. Além disso, foi percebido um aumento considerável no número de incidentes (chamados na empresa de “mensagens”) ao longo dos tempos, os quais foram associados, em um estudo preliminar, às diversas alterações legais recentes no escopo de atuação do software em questão. Como consequência, mais de 70% do tempo vinha sendo gasto em manutenção e a equipe estava sem tempo para trabalhar em outros projetos de inovação. Foi decidido então investir em ações com o objetivo de reduzir o esforço de manutenção.

Investir em mais qualidade é indispensável para se reduzir o esforço de manutenção, diminuir o índice de defeitos e consequentemente o de reclamações, melhorando assim a satisfação do cliente. Através do uso de ferramentas e técnicas de qualidade é possível realizar a identificação de oportunidades de melhoria e potenciais problemas promovendo assim o melhoramento contínuo. Entretanto, tentar resolver todos os problemas de uma vez pode não ser uma boa abordagem, afinal, conforme explica Lucinda, as organizações têm muito mais problemas do que recursos e energia para resolvê-los [9]. Mas com tantos produtos,

funcionalidades/produtos, tipos de problema e principalmente falta de recursos, por onde começar?

Este trabalho tem como objetivo propor e testar um sistema para identificação de oportunidades de melhorias e potenciais não conformidades utilizando como base de dados os incidentes reportados por clientes, promovendo assim a melhoria contínua. Para que este objetivo seja alcançado, foram definidos os seguintes objetivos específicos: a) Estudar ferramentas e técnicas de qualidade e melhoria contínua; b) Desenvolver o sistema; c) Utilizar o sistema na prática; d) Alisar os resultados obtidos. Este sistema foi utilizado pela equipe em estudo, para a diminuição do número de incidentes recebidos. Também foi realizada uma análise dos resultados da utilização do sistema na prática.

O texto segue como descrito. Na seção 2 é apresentada uma breve revisão bibliográfica. Já na seção 3 são apresentados os trabalhos relacionados. A seção 4 descreve a metodologia adotada para o desenvolvimento do trabalho. Na seção 5 são analisados os resultados obtidos com estudo de caso. As conclusões são descritas na seção 6, junto com a indicação dos trabalhos futuros previstos. A seção 7 apresenta as referências.

II. REVISÃO BIBLIOGRÁFICA

Esta seção resume aspectos dos temas Qualidade e algumas de suas ferramentas, alguns conceitos da NBR ISO 9000 referentes a melhoria contínua e satisfação do cliente, bem como o ciclo do processo PDCA, com o propósito de proporcionar ao leitor um contexto inicial motivador deste trabalho[1].

Vários conceitos da palavra qualidade foram utilizados ao longo dos anos. DEMING (1982) definiu qualidade como conformidade de um produto com as especificações técnicas que lhe foram atribuídas [5]. Já Campos (1992) define qualidade de acordo com as necessidades do cliente:

“Um produto ou serviço de qualidade é aquele que atende perfeitamente, de forma confiável, de forma aceitável, de forma segura e no tempo certo às necessidades do cliente. Portanto, em outros termos pode-se dizer: projeto perfeito, sem defeitos, baixo custo, segurança do cliente, entrega no prazo certo no local certo e na quantidade certa. O verdadeiro critério de boa qualidade é a preferência do consumidor”[4].

Observa-se que é fundamental entender, antes de qualquer coisa, que o conceito de qualidade depende da percepção de cada um, sendo, portanto, função da cultura do grupo que se considera [6].

Existem várias ferramentas de qualidade que auxiliam na melhoria de produtos, serviços e processos. Serão resumidas aqui algumas delas, as quais serão utilizadas neste trabalho.

A técnica dos cinco porquês busca identificar a causa raiz de um problema através de uma forma bem simples. Consiste em perguntar repetidamente o porquê daquele problema, sempre questionando a causa anterior até se atingir o nível considerado raiz, no qual não é mais possível determinar a causa [11].

Não é necessário que sejam exatamente cinco iterações, esse número pode ser menor ou maior. O importante é se chegar a causa raiz do problema. Também, eventualmente, pode haver desdobramentos na horizontal em casos onde se tem mais de uma causa para um porquê, por isso deve-se tomar cuidado para que não se perca o foco do problema original.

Scartezini ressalta que é importante ter em mente que esta é uma técnica limitada e que não substitui uma análise de qualidade detalhada [11]. Pessoas diferentes provavelmente chegarão a causas diferentes. Por isso é ideal que as perguntas sejam feitas com a participação de toda a equipe ou mais de uma pessoa pelo menos.

A estratificação é uma ferramenta que tem como função a busca das causas ou origens de um problema através da classificação dos dados em subgrupos baseados em características ou categorias [2]. Esse processo é ilustrado na figura 1 abaixo, onde as figuras geométricas no retângulo de cima representam os dados brutos e abaixo elas são classificadas em categorias, cada retângulo representando uma categoria.

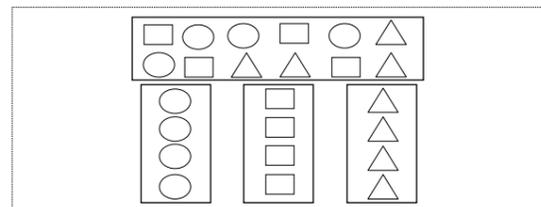


Figura 1. Ilustração do processo de estratificação

O diagrama de Pareto é uma técnica de priorização que auxilia a focar os esforços nas causas de problemas mais críticos mostrando a frequência dessas causas em ordem decrescente [12]. O princípio 80/20 de Pareto mostra que uma pequena parte das causas (aproximadamente 20%), que fica do lado esquerdo do diagrama, contribui para a maior parte dos efeitos observados (aproximadamente 80%) e a maior parte das causas (aproximadamente 80%), que fica do lado direito do diagrama, contribui muito pouco para os efeitos observados (aproximadamente 20%). A figura 2 apresenta um exemplo de utilização do diagrama de Pareto mostrando as causas de cem atrasos em um hospital em ordem decrescente.

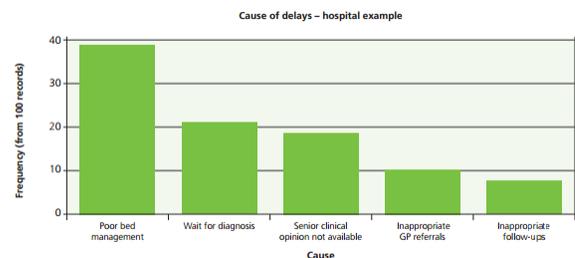


Figura 2. Frequência de causas de atraso em um hospital

Entretanto, Scartezini ressalta que o diagrama de Pareto define apenas o fator mais frequente na análise e não necessariamente o mais importante [11]. Conforme observa o

autor, os problemas mais frequentes nem sempre são os mais caros [11].

Logo, para uma empresa que busca reduzir seus custos, por exemplo, o defeito que mais gera reclamações pode não ser a prioridade número um, pois não é necessariamente o que gera o maior custo. Entretanto, para uma empresa que busca diminuir o número de reclamações, é mais relevante focar seus esforços nos defeitos que mais geram incidentes. Os problemas mais frequentes nem sempre são os que necessitam de mais atenção, deve-se sempre avaliar melhor a situação caso a caso.

Para uma melhor análise e priorização, é possível utilizar o diagrama de Pareto juntamente com a Matriz GUT. A Matriz GUT (Gravidade, Urgência e Tendência) é uma forma de priorização e tomadas de decisão baseado em medidas. As letras têm o seguinte significado, conforme descreve Scartezini [11]:

- G (gravidade): diz respeito ao impacto do problema sobre os processos, pessoas, resultados. Refere-se ao custo por deixar de tomar uma ação que poderia solucionar o problema;
- U (urgência): relaciona-se com o tempo disponível ou necessário para resolver o problema;
- T (tendência): diz respeito ao rumo ou propensão que o problema assumirá se nada for feito para eliminar o problema.

A filosofia do GUT é atribuir notas de 1 a 5 para cada uma das variáveis G, U e T dos problemas listados e tomar o produto destas notas como a criticidade do problema.

A NBR ISO 9000 define a melhoria continua como atividade recorrente para aumentar a capacidade de atender requisitos. Tem como objetivo aumentar a probabilidade de melhorar a satisfação dos clientes e de outras partes interessadas. A satisfação do cliente pode ser descrita como a sua percepção do grau no qual os seus requisitos foram atendidos. Ações para melhoria incluem desde a análise e avaliação da situação existente para identificar áreas para melhoria até a formalização das alterações caso as metas dessas alterações sejam atendidas. Após, os resultados são analisados para se determinar oportunidades adicionais de melhoria. Desta maneira, a melhoria passa a ser uma atividade continua (NBR ISO 9000).

O Ciclo PDCA é utilizado para o melhoramento contínuo da qualidade e consiste em percorrer continuamente as atividades de planejamento (Plan), execução (Do), verificação (Check) e ação corretiva (Act).

Barbosa (2010) define as atividades do ciclo PDCA como:

a) Planejamento (Plan): é estabelecer um plano que pode ser um cronograma, um gráfico ou um conjunto de padrões. As metas podem decorrer do plano ou vice versa.

b) Execução (Do): execução das tarefas exatamente como previstas no plano e coleta de dados para verificação do processo. Nessa etapa é essencial o treinamento no trabalho decorrente do plano.

c) Verificação (Check): a partir dos dados coletados na execução, compara-se a meta realizada com a planejada.

d) Atuação Corretiva (Act): Esta é a etapa onde o responsável pelo processo detectou desvios e atuará no sentido de fazer correções definitivas de tal modo que o problema nunca mais volte a ocorrer [3].

A figura 3 ilustra o processo de melhoramento contínuo do ciclo PDCA passando pelas quatro etapas (planejamento, execução, verificação e atuação corretiva) quantas vezes se julgar necessário.

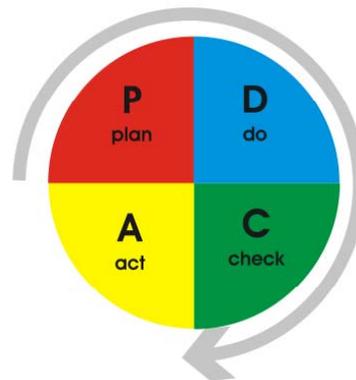


Figura 3. Ciclo da melhoria contínua ou Ciclo PDCA

III. TRABALHOS RELACIONADOS

Uma pesquisa foi feita para levantar e analisar trabalhos semelhantes com o propósito de buscar ideias e sugestões para complementar este trabalho. A partir dos resultados obtidos, dois dos mais relevantes para o trabalho atual serão brevemente comentados a seguir.

No primeiro trabalho relacionado [2], o autor também busca identificar problemas e melhorias utilizando informação de clientes, mas apesar dos resultados satisfatórios, o faz através de questionários com questões objetivas enviadas ao cliente. Em contraponto, a ideia apresentada neste trabalho tem como finalidade detectar potenciais problemas e melhorias por meio de uma análise textual, sendo aproveitado para este propósito, os incidentes já reportados pelos clientes.

O segundo trabalho [10], que mais se assemelha à ideia apresentada por este trabalho, busca reduzir o número de reclamações dos clientes em uma empresa do setor de bebidas, identificando problemas específicos e oportunidades de melhoria por meio da utilização das ferramentas de qualidade e os dados das reclamações dos clientes. São feitas estratificações nos dados coletados e as informações são apresentadas em diagramas de Pareto. A diferença para a ideia apresentada neste trabalho é que as estratificações e os diagramas de Pareto são feitos de forma manual e executados várias vezes para se chegar às informações desejadas. Por exemplo, primeiramente é realizada uma estratificação e cria-se um gráfico mostrando os produtos com mais reclamações. Ao se identificar o produto com o maior número de reclamações, os mesmos passos são repetidos para identificar os tipos de problemas que mais ocorrem naquele produto. Esse

processo é repetido para se identificar em que turno isso mais aconteceu, em qual filial, etc. Na ideia apresentada neste trabalho, todos esses dados (produto, tipo de problema, componente ou setor) são carregados de uma só vez e em alguns clicks é possível observar todas as informações através da aplicação desenvolvida. Outra diferença é que toda a informação obtida anteriormente faz parte do produto com maior número de problemas. E se quisermos ver toda essa informação dos outros produtos? Não é possível ter uma visão geral dos problemas. É preciso repetir todo esse processo novamente para cada produto. Em contraponto, na ideia apresentada neste trabalho, as informações podem ser observadas de uma maneira geral com opções de filtro para tornar a informação mais específica.

IV. METODOLOGIA

Neste capítulo é descrita a metodologia desenvolvida para atender às necessidades do estudo de caso abordado neste trabalho.

Foi identificada em uma das equipes da empresa, que é objeto deste estudo, a necessidade de reduzir o esforço de manutenção e o número de incidentes. Através de um histograma, foi possível verificar um aumento considerável no número de incidentes ao longo dos anos. Para se chegar ao objetivo desejado de redução da entrada de novos incidentes, foi utilizado o estudo de caso para acompanhar a atividade diária da equipe, analisar os aspectos de qualidade, entender como é realizada a priorização dos itens de qualidade, quais ações são tomadas para diminuição dos incidentes e manutenção, etc., possibilitando assim propostas de melhoria.

A. Estudo de Caso

Após acompanhar por um período a atividade diária da equipe, foi possível perceber que cada vez mais era necessário um esforço extra da mesma para suprir as demandas de trabalho. Para tentar amenizar essa questão, foi criada uma política de eficiência, de tentar fazer mais com menos, aprimorando os processos e removendo tarefas que não trariam tanto retorno, bem como alguns tipos de testes e verificações. Apesar do número de incidentes e manutenção não diminuir, essa ação fez com que a equipe conseguisse lidar melhor com a carga de trabalho e começasse a pensar em ações para diminuir a entrada de problemas, o que traria uma melhor contribuição para diminuição da carga de trabalho.

A equipe decidiu melhorar a qualidade do software, cobrindo mais o código fonte com testes automatizados. Entretanto, essa decisão foi tomada sem a realização de uma análise profunda dos incidentes. Foi possível observar uma melhora, pois sempre que uma alteração tinha que ser feita no software, o teste automatizado criado anteriormente possibilitava um grau maior de confiança para o desenvolvedor, que poderia verificar se sua alteração não comprometia casos que já estavam funcionando. Mesmo assim esta opção não era adequada o bastante, pois conforme será descrito adiante, problemas de código não representavam sozinhos a grande maioria dos problemas. Além disso, era inviável fazer testes unitários em todos os produtos/funcionalidades e não havia uma priorização

adequada, ou seja, não eram selecionados primeiro os produtos/funcionalidades mais relevantes ou os mais problemáticos para se realizar testes automatizados e sim quando era necessário fazer alguma alteração naquele produto/funcionalidade, era criado um teste automatizado.

Para a análise dos incidentes, a empresa conta com um processo de classificação dos mesmos no momento em que são fechados. Entretanto, essa categorização é muito genérica e utilizada pela alta gerência, não suprimindo a necessidade a nível de equipe.

B. Ciclo PDCA

Através do ciclo PDCA, foi definido como meta reduzir o esforço de manutenção. Como a maior parte das atividades de manutenção estava relacionada a resolução de incidentes de clientes, foi decidido focar principalmente em reduzir esse número. Por consequência, diminuir os problemas reportados também trará uma melhor satisfação ao cliente.

Etapa PLAN: após uma análise superficial através de uma amostra de 510 mensagens, de um período de um ano, do histórico de incidentes, foi constatado que não eram apenas erros de código o principal motivo pelo qual os clientes abriam mensagens, ao contrário, estes erros não representavam nem a metade daquela amostra conforme mostra a figura 4. Ou seja, a melhora nos testes de software iria atender a apenas uma parte do problema dos incidentes, além de não haver a priorização dos produtos/funcionalidades. Então se decidiu realizar uma análise mais profunda dos incidentes novos e antigos.

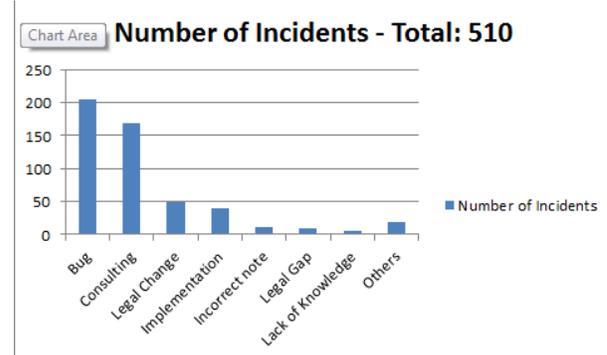


Figura 4. Frequência das causas dos incidentes

Etapa DO: a equipe se reuniu em uma reunião de brainstorming para definir os produtos/funcionalidades e tipos de problema em que seriam categorizados os incidentes.

Através do método de estratificação, as mensagens foram classificadas nos grupos determinados de produto/funcionalidade e problema e gravadas em uma planilha eletrônica. Selecionar o produto/funcionalidade era uma tarefa mais simples, pois na maioria das vezes, estava claro em qual produto/funcionalidade o problema ocorria. Porém para identificar claramente o problema era necessário uma análise mais aprofundada. A técnica dos 5 porquês auxiliou a identificar o tipo de problema. Realizando a iteração da técnica algumas vezes, questionando inicialmente

o motivo pela qual o cliente abriu a mensagem, auxiliou a se chegar em uma das categorias definidas. Exemplo:

- 1ª iteração: Por que o cliente abriu o incidente? Porque não estava conseguindo executar o programa A.
- 2ª iteração: Por que não estava conseguindo executar o programa A? Porque o programa parava inesperadamente após alguns segundos.
- 3ª iteração: Por que o programa parava inesperadamente após alguns segundos? Porque o método A não está corretamente implementado.
- 4ª iteração: Por que o método A não está corretamente implementado? Porque o cliente não instalou a nota de correção corretamente (erro de implementação). Ou também nesse caso poderia ser porque o método A não foi desenvolvido corretamente (bug).

Algumas informações adicionais que podem ser úteis, como data da criação das mensagens e componentes do software, também foram adicionadas à planilha.

Com essas mensagens classificadas foram gerados gráficos semelhantes a diagramas de Pareto para produtos/funcionalidades e tipos de problema como por exemplo é mostrado nas figuras 5 e 6.

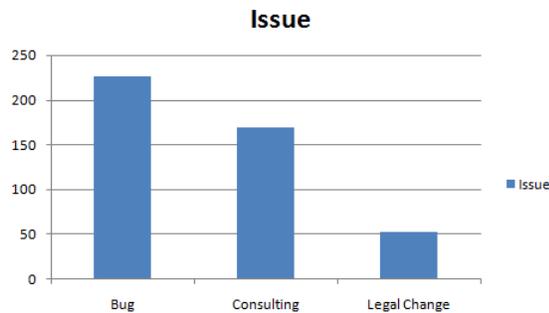


Figura 5. Exemplo que mostra os tipos de problemas mais identificados nos incidentes

O gráfico da figura 5 contém apenas os 3 problemas principais encontrados. É possível observar que os 3 problemas mais frequentes eram, em ordem decrescente, erros de código, consultoria e alterações legais.

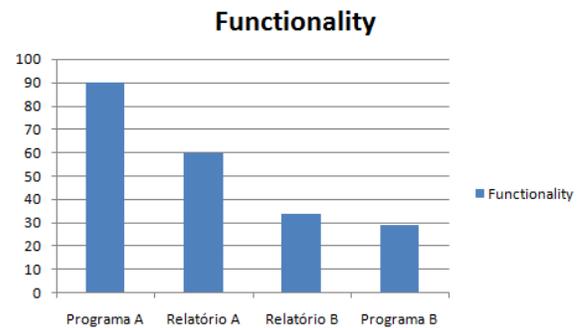


Figura 6. Exemplo que mostra os produtos/funcionalidades mais identificados nos incidentes

O gráfico da figura 6 apresenta apenas os 4 produtos/funcionalidades com maiores problemas em ordem decrescente. Podemos ver que “programa A” (os nomes foram alterados) é o produto/funcionalidade mais afetado e o “programa B” é o quarto produto/funcionalidade com mais problemas.

Etapa CHECK: apesar dos resultados satisfatórios, com a identificação da frequência de produtos/funcionalidades e problemas em ordem decrescente, era selecionado, por exemplo, o produto/funcionalidade mais afetado para ser resolvido, mas ainda assim não se sabia qual o maior problema daquele produto/funcionalidade. E também foi observado o problema contrário, com a decisão de resolver um determinado problema, quando este se mostrava muito abrangente, tornava-se necessário definir um produto/funcionalidade para priorizar. Outras questões também foram levantadas, como por exemplo, qual o país mais afetado? Qual o componente mais atingido? Setor público ou setor privado deve ser mais analisado?

Etapa ACT: foi decidido então desenvolver uma ferramenta que faz o carregamento da planilha e a geração do *dashboard* interativo em que se cruza as informações de funcionalidade e problema. Então é possível selecionar a funcionalidade/produto e ver os problemas que mais acontecem naquela funcionalidade/produto ou selecionar o problema e verificar a funcionalidade/produto mais afetada por aquele problema. Além disso, é possível fazer filtros de data, para ver períodos onde mais se encontraram problemas de uma determinada funcionalidade/produto ou de um determinado problema e ainda selecionar qual o componente a qual pertence aquele problema.

C. Ciclo de Desenvolvimento da Ferramenta

A seguir são descritas as etapas do desenvolvimento da ferramenta.

1) Requisitos e Modelagem do Sistema

Para a ferramenta a desenvolver foram definidos como requisitos os aspectos descritos a seguir. O sistema deveria carregar a planilha já criada e preenchida com os dados desejados, e gerar os *dashboard* dinâmicos, com opções de filtro de funcionalidade/produto e tipo de problema, além de filtros de data e componente. Deveria ser de fácil acesso e em poucos clicks o usuário deveria conseguir gerar os

dashboards, ou seja, ter uma boa usabilidade. Para melhorar a usabilidade, alguns botões rápidos para mostrar os resultados do mês, do ano atual e para voltar a mostrar todos os resultados podem ser criados.

Foi decidido então criar uma aplicação web e disponibilizá-la na intranet onde bastava uma URL para qualquer usuário ter acesso. Foi utilizado um módulo já pronto da empresa que identifica o usuário automaticamente através da tecnologia *Single Sign-on*, logo não era necessário criar suporte para o cadastramento dos usuários, agilizando assim o processo de utilização do sistema.

Pelo maior conhecimento do desenvolvedor e por ser uma linguagem de rápido desenvolvimento, em concordância com a equipe, foi decidido utilizar a linguagem PHP, que é voltada para aplicações web. A linguagem de programação foi utilizada para criação e leitura das planilhas, geração dos arquivos em formato XML e para as opções de filtro de dados. A interface foi criada utilizando HTML juntamente com CSS e JavaScript, este último utilizado principalmente para os filtros assíncronos. Foi utilizado como base de dados arquivos XML, pois seriam poucos os dados a serem gravados e também evitaria o custo de configurar um banco de dados. Para visualização dos gráficos dinâmicos, foram criados arquivos flash gerados pela ferramenta Xcelsius. A ferramenta Xcelsius permite ao usuário criar versões flash de planilhas em apresentações gráficas com interatividade em tempo real e com várias formas de conectividade para busca dos dados como XML, banco de dados, variáveis flash, etc.

2) Criação da Planilha de Incidentes e Opções de Categorização Pré-Definidas

Através de uma reunião de brainstorming com toda a equipe, foi definida a lista de funcionalidade/produtos e causas de problema. As duas listas foram adicionadas a uma aba na planilha chamada de configuração, onde poderiam ser utilizadas como opções pré-definidas na hora de categorizar as mensagens que são guardadas em outra aba específica para as mensagens. Pode-se observar a estrutura da aba de configuração na figura 7 com a lista de funcionalidade/produtos e a lista de causas com uma pequena descrição ao lado para auxiliar na hora da categorização.

* Functionalities list		* Problem cause list	
Reports/Functionalities		Problem Cause	Description
Programa 1		Bug	Needs code correction, customer
Programa 2		Consulting	Doubts about how system works
Programa 3		Design	Technical issue (architecture)
Programa 4		Implementation	Customer had not implemented
Programa 5		Incorrect note	Problem in the note/note document
Programa 6		Lack of Knowledge	Doubts about how business works
Programa 7		Legal Change	Requesting or piloting legal changes
Programa 8		Legal Gap	Legal gap in software, requires
Programa 9		Legal Grey Area	Legal documentation is not clear
Programa 10		New Requirement	Request of new requirements
Programa 11		No Fill	Not relevant miscellaneous message
Programa 12		Out of IM	
Programa 13		Solved by the customer	
Programa 14			
Programa 15			
Programa 16			
Programa 17			
Programa 18			
Programa 19			
Programa 20			
Programa 21			

Figura 7. Aba de configuração da planilha

A aba ao lado, de mensagens, é onde são guardadas as mensagens e suas respectivas categorizações. A figura 8 ilustra esta aba com um exemplo de categorização.

Sent Date	Msg No	Component	Report/Functionality	Problem Cause
29/04/2013	398472	PY-PT-PS	Programa 1	Legal Change

Figura 8. Aba de mensagens da planilha

Os dados da aba de mensagens serão utilizados pela aplicação para gerar os gráficos, a aba de configuração é utilizada apenas para auxiliar na inclusão das mensagens.

3) Desenvolvimento da Interface da Aplicação

A interface foi dividida em 3 abas principais que foram criadas utilizando o “Accordion” da biblioteca JavaScript “Jquery”, que gera um efeito sanfona. A primeira é aba onde são apresentados os *dashboards* que mostram os tipos de problemas que mais ocorrem com opção de filtro por funcionalidade/produto e também o oposto, as funcionalidades/produtos mais problemáticas com opção de filtro por causa do problema conforme figura 9. Observa-se na figura, que à esquerda é exibido o gráfico com a opção selecionada de visualização de problemas e também um filtro de funcionalidades/produtos mais ao centro. À direita estão localizados os links rápidos de tempo (“tudo”, “mês corrente” e “ano corrente”), filtros por componente e data. Também há um botão para atualizar o *dashboard* em caso de alteração manual dos arquivos XML ou de alguns problemas com o browser, onde o gráfico pode não se alterar mesmo fazendo um novo upload.

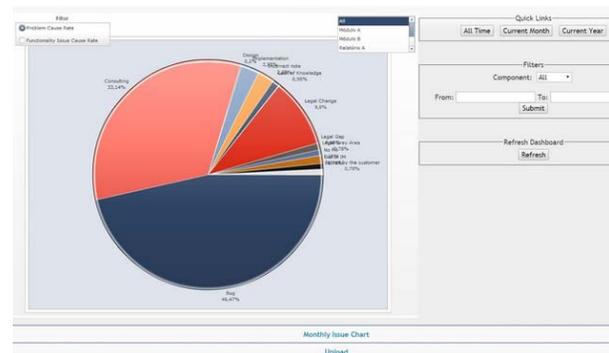


Figura 9. Primeira aba da aplicação, utilizando a opção de visualização de causas

A figura 10 mostra a mesma tela da figura 9, porém a opção de funcionalidades/produtos está selecionada, então um

gráfico de barras com as funcionalidades/produtos é mostrado no lugar do gráfico de pizza com os problemas. E no centro, também é possível ver a opção de filtro por tipo de problema e não mais por funcionalidade/produto.

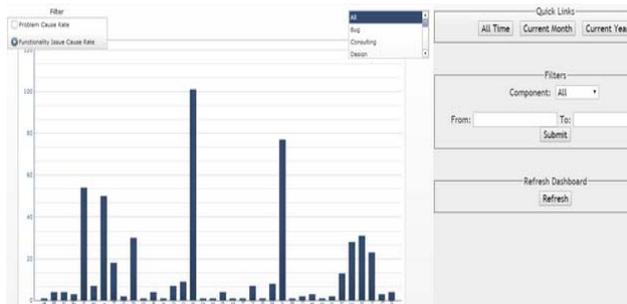


Figura 10. Primeira aba da aplicação utilizando a opção de visualização de funcionalidades

A segunda aba foi desenvolvida para haver um histórico das mensagens ao longo do ano, com opções de filtro por ano (Figura 11). Observa-se na figura 11, por exemplo, que há um pico de mensagens de 2013 em Fevereiro, enquanto Setembro e Dezembro são os meses que a menor quantidade de mensagens foi resolvida.



Figura 11. Segunda aba da aplicação

A terceira e última aba (Figura 12) possibilita realizar a operação de upload da planilha de ocorrências, além de conter o link para o *template* da planilha e um tutorial.

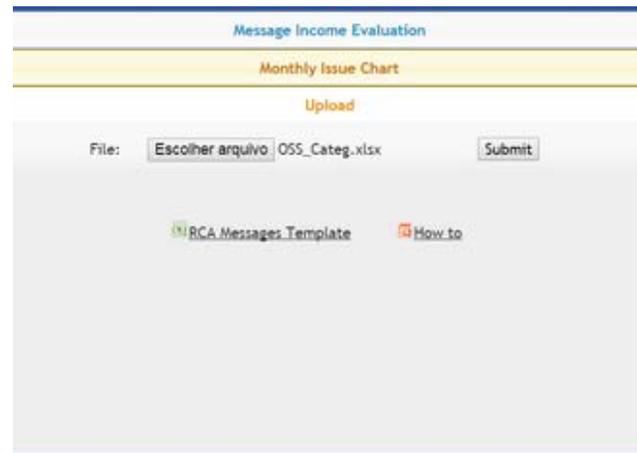


Figura 12. Terceira aba da aplicação

4) Desenvolvimento do Módulo de Carregamento da Planilha e Geração dos Arquivos de Dados para Leitura

O carregamento da planilha (arquivo “XLS” ou “XLSX”) é realizado na terceira aba da aplicação. Assim que o usuário realiza o carregamento da planilha, o módulo de leitura faz o cálculo das linhas e gera o resultado desse cálculo em arquivos XML, bem como uma cópia da planilha para uso nos filtros. Tanto os arquivos XML, quanto a cópia da planilha, são nomeadas com a identificação do usuário e salvas em uma pasta específica onde a aplicação está rodando. Portanto vários usuários podem fazer o carregamento dos dados e a utilização do sistema simultaneamente, pois cada um terá seus dados separados pela identificação.

5) Desenvolvimento dos Dashboards Interativos

Através da ferramenta Xcelsius, foi desenvolvido um *dashboard* para apresentar de forma clara as funcionalidades/produtos mais problemáticas, semelhante a um gráfico de Pareto. Um segundo *dashboard* apresenta os tipos de problemas mais encontrados com opção de filtro por funcionalidade/produto. Ambos fazem a leitura dos dados nos arquivos XML gerados na etapa de carregamento da planilha. Os arquivos de dados são selecionados de acordo com a identificação do usuário.

6) Desenvolvimento de Opções de Filtro por Componente e Data

Foram criados filtros por componente e por data, além de links rápidos para filtrar os resultados do mês ou ano corrente. Os filtros, quando utilizados, geram uma requisição POST e passam os campos para outro método que gera um arquivo XML temporário com os dados filtrados conforme selecionado. Foi utilizada também a tecnologia AJAX, ou seja, não é atualizada toda a página quando o botão é utilizado e sim apenas o *dashboard* melhorando assim a usabilidade da aplicação.

7) Definição do Processo de Utilização do Sistema

Sempre que a equipe julgar necessário, é possível atualizar as categorias de funcionalidade/produto e o tipo de problema. Assim como toda vez que um incidente novo é resolvido, deve ser feita sua classificação, para evitar que muitos incidentes

não classificados se acumulem, o que pode gerar uma bola de neve e dificultar a utilização do sistema no futuro.

Outras equipes avaliaram de forma positiva a ideia e também iniciaram a utilização do sistema. Por isso foi criado um tutorial descrevendo os passos de utilização do sistema. Em resumo, são os seguintes:

1. Definir as categorias de funcionalidades/produtos e problemas e inserir na planilha modelo. Na primeira utilização, recomenda-se fazer uma reunião de brainstorming com a equipe;
2. Coletar uma quantidade considerável de mensagens (em torno de 100 pelo menos) e inserir na planilha modelo;
3. Utilizar o método de estratificação com a ajuda dos 5 porquês para classificar as mensagens inserindo as classificações nas respectivas colunas;
4. Acessar o link da aplicação e carregar a planilha para que sejam gerados os gráficos;
5. Utilização da Matriz GUT (caso necessário).

A figura 13 ilustra, através de um fluxograma, os passos do sistema:

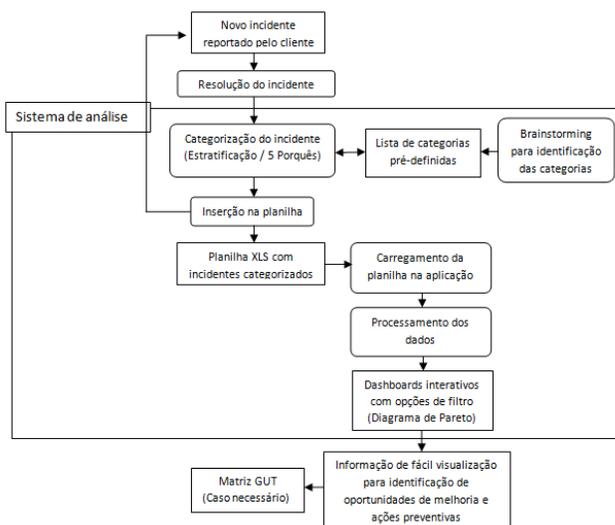


Figura 13. Fluxograma do sistema

Como neste caso, a intenção era apenas reduzir o número de mensagens, não foi levado em conta o grau de criticidade dos potenciais problemas. Caso a equipe julgue necessário, é possível adicionar o passo de análise com a Matriz GUT (passo 5), para identificar o grau de criticidade dos potenciais problemas atribuindo notas para cada um dos três aspectos principais: Gravidade, Urgência e Tendência.

Com o sistema definido, este pode ser utilizado na etapa PLAN de outros ciclos PDCA para identificação de uma oportunidade de melhoria ou potencial não conformidade, iniciando outro ciclo de melhoria contínua.

V. ANÁLISE DOS RESULTADOS

A seguir serão descritas atividades relacionadas à análise preliminar do sistema desenvolvido. Estas análises possuem como finalidade, ampliar a percepção de utilidade e adequação do sistema, bem como proporcionar elementos para análise que permitam a identificação de melhorias possíveis. Também é descrito os resultados de algumas ações tomadas com o auxílio do sistema.

Enquanto a aplicação era desenvolvida, a equipe categorizou incidentes equivalente a um ano, além de categorizar os novos incidentes. Após a conclusão do desenvolvimento da aplicação, o sistema proposto foi utilizado com os dados coletados e devidamente categorizados.

Através da utilização do sistema, foi possível identificar pontos críticos para serem melhorados ou melhor analisados. Na figura a seguir (Figura 14), podem ser vistos os resultados do sistema proposto em relação as funcionalidades/produtos a serem analisadas.

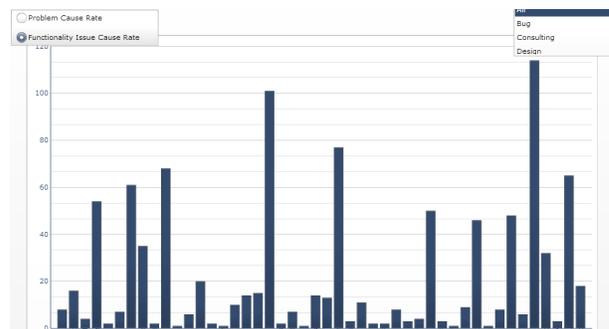


Figura 14. Funcionalidades/produtos com mais problemas

Um total de 998 incidentes foram classificados com 46 diferentes funcionalidades/produtos. Seleccionando as 11 funcionalidades/produtos com maior quantidade de problemas, o que equivale a aproximadamente 24% do total de funcionalidades/produtos, obtém-se os números descritos na tabela 1.

Tabela 1. Funcionalidades com mais incidentes

Funcionalidade	Nº de incidentes
Programa 1	114
Programa 2	101
Programa 3	77
Programa 4	68
Programa 5	65
Programa 6	61
Programa 7	54
Programa 8	50
Programa 9	48
Programa 10	46
Programa 11	32
Total	751 (~75% do total de incidentes)

Ou seja, cerca de 75% dos problemas estão concentrados em aproximadamente 24% das funcionalidades. Observa-se

que o princípio de Pareto, citado anteriormente, pode ser considerado como presente neste cálculo, pois os números chegam muito próximos a 80/20 proposto por Pareto.

Na figura 15 podemos visualizar de maneira geral os problemas mais frequentes dos incidentes analisados.

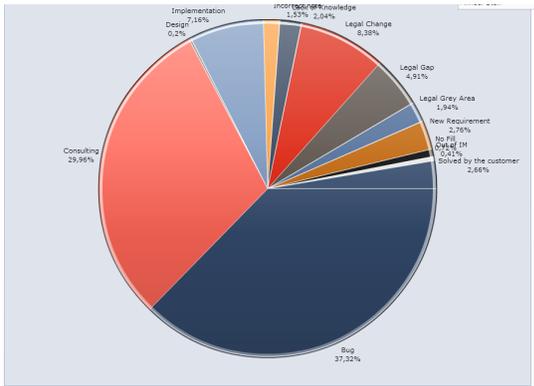


Figura 15. Causas mais frequentes

É possível identificar que somando as 5 causas mais frequentes das 13 diferentes causas, se obtém um valor de 82,8% do total das causas. Ou seja, focando nessas 5 causas principais, pode ser atendida a maioria dos problemas. Analisando apenas as duas mais frequentes (Bug e Consulting) observa-se 67,28% das causas, o que se constitui em um aspecto relevante.

Porém com essa visão geral das causas, volta-se ao problema mencionado anteriormente, atacar uma causa pode ser muito abrangente. Por isso, foi selecionado nas opções de filtro as duas funcionalidades com maior quantidade de problemas identificados anteriormente, identificadas aqui como programa 1 e programa 2.

Na figura 16 pode ser visto o resultado desse filtro para o programa 1:

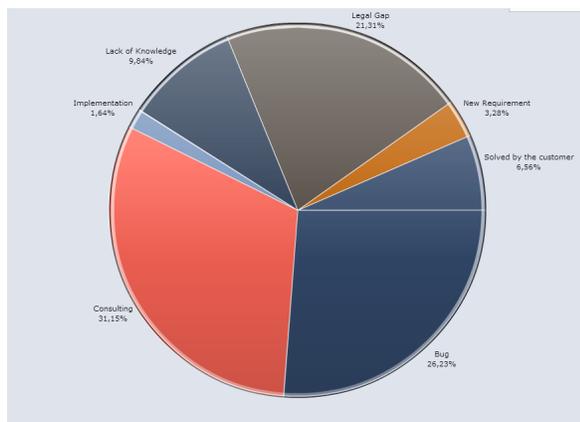


Figura 16. Causas mais frequentes do Programa 1

É possível identificar vários tipos de causas de problema nessa funcionalidade, mais as principais causas são consultoria, problemas de código e lacunas que o programa não cobria. É possível perceber que, ao filtrar o gráfico para mostrar apenas causas do Programa 1, os erros de código já

deixaram de ser a causa principal dos problemas como era na visão geral conforme mostrado na figura 20, passando agora a ser a consultoria, a causa mais frequente. A consultoria, no entendimento da equipe, engloba principalmente as dificuldades que o cliente tem de utilização do programa, o que pode ser sinal de software não intuitivo. Logo, o resultado dessa análise serviu de justificativa para ser realizada uma reconstrução no programa, com novos testes automatizados, código fonte totalmente alterado utilizando paradigma orientado a objetos, forte atenção na documentação para o usuário final, etc. Para o problema das lacunas que o programa não cobria, foi realizada uma reunião com as partes interessadas para levantar novamente os requisitos e criar um projeto detalhado do programa.

O mesmo processo foi realizado para o Programa 2, a segunda funcionalidade com mais problemas segundo o gráfico de funcionalidades/produtos apresentado anteriormente. No programa 2, foi identificado como principal problema os erros de código (bugs), conforme mostra a figura 17.

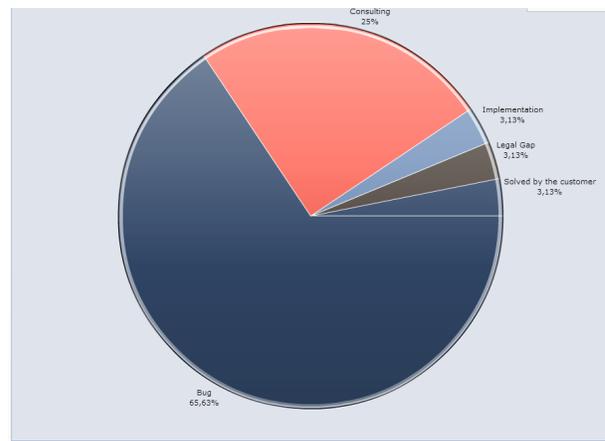


Figura 17. Causas mais frequentes do Programa 2

Para esta funcionalidade, como a maior parte dos problemas está relacionado a erros de código, as ações tomadas, após um planejamento, foram para melhorar os testes, criando vários testes automatizados e uma bateria geral de testes foi realizada.

A. Resultados das Ações Tomadas

Como resultado da ação tomada referente aos testes unitários do Programa 2, houve uma redução de 60% no número de mensagens após uma alteração legal feita no programa, comparado à média de mensagens que chegavam após outras alterações legais do passado. Considerando que esta é a segunda funcionalidade/produto com maior número de mensagens abertas, é possível que haja uma redução considerável no total de mensagens.

Não é possível afirmar que essa redução ocorreu apenas pelo fato das ações tomadas, pois muitas variáveis estão envolvidas, como por exemplo, complexidade da alteração legal, número de clientes que utilizaram o programa após essa

alteração legal, quem a desenvolveu, etc. Mas é bastante provável que os testes unitários providenciam uma confiança ao desenvolvedor, que pode verificar se sua alteração não compromete casos que já estavam funcionando.

Em relação ao refatoramento do relatório do Programa 1, foi desenvolvido utilizando uma metodologia chamada Design Thinking, que aproxima o cliente da equipe de desenvolvimento. Como o refatoramento exige mais tempo, ainda está em fase de testes, entretanto já foi possível observar vários feedbacks positivos dos clientes em relação à usabilidade através das maquetes de telas que foram criadas durante o processo. Visto que um dos maiores problemas do Programa 1 era consultoria, uma melhor usabilidade pode reduzir bastante o número de mensagens. Uma especialista em documentação também trabalhou junto à todo processo de desenvolvimento, para garantir que os clientes tenham acesso à uma informação clara da utilização do programa.

VI. CONCLUSÃO

Investir em qualidade é indispensável para uma organização se manter competitiva, portanto esforços neste sentido não devem ser considerados gastos desnecessários. A melhoria contínua, quando promovida, aumenta a probabilidade de melhorar a qualidade do processo, qualidade do produto e a satisfação dos clientes. A etapa de priorização das melhorias e ações preventivas é fundamental no processo de melhoria contínua pois garante que as questões mais relevantes sejam analisadas primeiro, trazendo um maior retorno para a organização.

A equipe que participou do estudo de caso estava lidando com uma situação difícil pois o número de incidentes reportados pelos clientes estava crescendo de forma expressiva, devido a diversos fatores. O sistema proposto, composto por ferramentas de qualidade e aplicação, permitiu a equipe realizar uma análise mais profunda dos motivos pelos quais os clientes estavam reportando problemas, identificar melhorias e priorizá-las, focando assim seus esforços nas questões que mais eram alvo de reclamações dos clientes. Com as ações focadas nesses pontos críticos, acredita-se ser possível obter um retorno positivo mais rapidamente. A categorização das mensagens passou a fazer parte da rotina da equipe que poderá identificar outras oportunidades melhorias quando achar necessário. Por ser um sistema com processos genéricos e não específicos para área de software, pode ser utilizado por empresas de outros nichos de mercado.

Na tabela 2 é citado alguns pontos positivos e negativos identificados no trabalho:

Tabela 2. Pontos positivos e negativos

Pontos Positivos	Pontos Negativos
Possibilitou que equipe em estudo identificasse oportunidades de melhorias com foco no melhor ROI	Demanda esforço para o levantamento das categorias e classificação dos incidentes
Ações tomadas com resultados positivos	Pessoas diferentes podem chegar a classificações diferentes

É importante salientar que a ideia proposta com a Análise de Pareto foi utilizada para se obter uma priorização quantitativa, ou seja, que busca identificar os problemas que mais ocorrem, com o propósito de identificar ações para diminuir o número de recorrências. É possível que o problema que mais ocorre, não seja aquele que deve ser priorizado. Devem ser levados em conta vários outros fatores, tais como o número de clientes que utilizam aquela funcionalidade, a criticidade do problema, o quão complexa é sua resolução, entre outras, de acordo com os objetivos da empresa. A ideia proposta serve de apoio para tomada de decisão e deve ser utilizado em conjunto com outras técnicas.

A. Trabalhos Futuros

Algumas melhorias no sistema podem ser realizadas em trabalhos futuros. O processamento de linguagem natural (PLN) pode ser utilizado na etapa de estratificação e categorização das mensagens. Este recurso poderia ser utilizado para identificar padrões textuais e sugerir possíveis categorias para o usuário selecionar ou até mesmo selecioná-las automaticamente, agilizando o processo de categorização. Outra proposta é integrar os processos apresentados com o software que gerencia os chamados de clientes já utilizado pela empresa, agilizando todo o procedimento. Essa integração, juntamente com o PLN, pode gerar os gráficos em um processo totalmente automatizado, tendo que ser definido e configurado apenas a lista de categorias.

O processo de integração com o software que gerencia os chamados de clientes, já utilizado pela empresa, está em fase de desenvolvimento. As categorias estão sendo registradas no próprio software eliminando assim a planilha. Os gráficos também estão sendo integrados, sendo gerados no próprio software. Essa iniciativa busca diminuir o esforço de registrar os dados na planilha e utilização de mais uma aplicação.

Referências

- [1] Associação Brasileira de Normas Técnicas (ABNT). NBR ISO 9000. Rio de Janeiro, 2000.
- [2] ALVES, F.F., SANTOS, P.H. da C. Qualidade Total na Prestação de Serviços. 2010.
- [3] BARBOSA, E. F. Sete Ferramentas do Controle de Qualidade. Belo Horizonte, 2010.
- [4] CAMPOS, V. F. Controle da qualidade total (no estilo Japonês). Belo Horizonte: Fundação Christiano Ottoni, 1992.
- [5] DEMING, W.E. Quality, productivity, and competitive position, 1982 sob a ótica GOMES P. J. A evolução do conceito de qualidade: dos bens manufacturados aos serviços de informação. Cadernos Bad 2, 2004.
- [6] GRIFO, Equipe. Iniciando os conceitos de qualidade total. 6. ed.São Paulo: Pioneira, 1998.
- [7] KALINOWSKI, M., MENDES, E., TRAVASSOS, G.H. An Industry Ready Defect. 2014.
- [8] LESZAK, M., PERRY, D. E., STOLL, D. A case study in root cause defect analysis. 2000.
- [9] LUCINDA M.A. Qualidade - Fundamentos e Práticas. ed. Brasport. 2010.
- [10] PESSOA, G. A., SILVA, M. J. N. da. Aplicação do PDCA: redução do índice de reclamações de clientes. 2008
- [11] SCARTEZINI, L.M.B. Análise e Melhoria de Processos. 2009.
- [12] TAYLOR, J. R. A. The Handbook of Quality and Service Improvement Tools. NHS Institute for Innovation and Improvement. 2010.