

A IMPORTÂNCIA DA ENGENHARIA DE TESTES PARA A GARANTIA DA QUALIDADE DE SOFTWARE

THE TEST ENGINEERING IMPORTANCE FOR SOFTWARE QUALITY ASSURANCE

AZEVEDO, Livyson Saymon Leão¹; SILVA, Rafael Bezerra Correia da²
^{1,2}Faculdade dos Guararapes – FG, Recife, Pernambuco, Brasil
livyson@gmail.com; rafael.bcsilva@gmail.com

Resumo

O objetivo deste artigo é destacar a influência que a engenharia de testes tem sobre o processo de garantia da qualidade, descrevendo como ocorre toda a parte de planejamento, análise, execução, coleta e tomada de ação no processo de testes, destacando algumas das diversas formas de incluir a área na produção de software. Esse trabalho também aborda a hierarquia, cargos e responsabilidades que as empresas de desenvolvimento de sistemas informatizados adotam para a área de engenharia de testes, assim como os tipos de testes que são abordados por alguns projetos. O artigo ainda traz a utilização do Ciclo PDCA, 6 Sigma e Método DMAIC nos processos de produção dos softwares.

Palavras-chave: qualidade; software; testes.

Abstract

The purpose of this article is to highlight the influence that test engineering have on the process of quality assurance, describing how is all part of planning, analysis, execution, collecting and taking action in the testing process, highlighting some of the different ways to include the area in the production of software. This work also addresses the hierarchy, roles and responsibilities that companies developing computerized systems to adopt the test engineering, well as the types of tests that are adopted by some projects. The article also brings the use of the PDCA cycle, 6 Sigma and DMAIC Method in the production processes of the software.

Keywords: quality; software; tests.

1 INTRODUÇÃO

O uso de alguns princípios básicos da qualidade já era notável desde a antiguidade, quando artesãos tentavam atender as expectativas dos compradores através do produto construído. Naquela época, os mesmos não tinham noção do conceito da palavra “qualidade”, mas de alguma forma já sentiam a necessidade da utilização de processos e ferramentas que os guiassem na fase de produção.

Entre o fim do século XVIII e o início do século XIX, as indústrias começaram a surgir e com elas, o processo de padronização dos produtos. Era o início da utilização da qualidade no processo de produção.

Nos anos 70, a informática passa a mudar todos os ramos do conhecimento humano, exigindo das indústrias uma atualização que garantisse a utilização de equipamentos modernos e sofisticados. Essas mudanças não afetaram apenas os clientes finais desses dispositivos recém-inventados, mas também as organizações que os construíam. As empresas de tecnologia começaram a sentir a necessidade de padronizar o processo de construção, estimulando uma maior produção que atendesse à demanda do mercado, satisfazendo os funcionários envolvidos nos projetos e garantindo a qualidade dos produtos que estavam sendo fabricados.

No início dos anos 80, essa preocupação só aumentou. Os periféricos computacionais davam espaço para os softwares, que tinham sobre eles toda a responsabilidade de gerenciar inúmeras tarefas. Os softwares ganharam confiança e começaram a ser implantados em sistemas complexos, que envolviam cada vez mais vidas e estruturas, tais como os sistemas hospitalares e os sistemas para aeronaves.

Devido à complexidade dos softwares, era necessária a implantação da gestão da qualidade, que tinha como objetivo garantir que os processos de produção estavam sendo executados conforme o planejado. Mas só a

constância de propósitos dos processos não garantiria a qualidade do produto, para isso, a engenharia de testes foi acoplada, para investigar os softwares a fim de fornecer um parecer sobre a qualidade do produto em si.

Com o objetivo de entender como a engenharia de testes atua no processo de produção de software, garantindo a eficiência dos produtos desenvolvidos, de forma a atender às necessidades dos clientes, o estudo deste projeto de pesquisa analítica, bibliográfica e documental se baseia na seguinte questão: Qual a importância da Engenharia de Testes para a Garantia da Qualidade de Software?

2 METODOLOGIA

Um trabalho de pesquisa e produção científica tem como objetivo responder a algum questionamento. Para chegar à resposta desejada, é necessário utilizar o poder da análise crítica, avaliando dados e o objeto de estudo que contribuem para o desenvolvimento do que está sendo produzido. Para ter embasamento, dados e uma quantidade significativa de informações, o pesquisador precisará ter ou adquirir o hábito da leitura, sabendo interpretar os textos.

O objetivo da metodologia é chegar a uma resposta para a dúvida apresentada, dividindo o problema em várias partes, servindo assim como base para o projeto de pesquisa.

Foram realizadas pesquisas bibliográficas e documentais, que serviram de base para o entendimento da forma mais coerente de execução da qualidade e da engenharia de testes em uma empresa de desenvolvimento de software. A consulta à internet foi realizada em diversos momentos, sendo essa a maior fonte de contribuição de informações para o projeto.

3 FUNDAMENTAÇÃO TEÓRICA

O mercado de tecnologia é considerado um dos mais exigentes, devido ao fato de

trabalhar com projetos de grande complexidade, que muitas das vezes envolvem vidas e estruturas físicas. Erros em sistemas causam prejuízos significativos e a reputação da empresa pode nunca mais ser reerguida. É necessário um trabalho minucioso e um processo rígido durante toda a fase de produção.

De acordo com o COBIT 4.7 (2005), um evento ou uma condição que seja considerada como incerta, se ocorrer e quando ocorrer, gerará um efeito positivo ou negativo no projeto que está desenvolvendo um produto.

Esses riscos devem ser trabalhados por uma equipe que garanta a estabilidade do projeto e do produto que está sendo desenvolvido. Algumas normas foram criadas para que possam auxiliar nesse processo de garantia da qualidade. Essas normas são aprimoradas das experiências de profissionais especializados que debatem o assunto em fóruns, com base em implantações e consultorias realizadas em diversas empresas no mundo inteiro.

“Para que se tenha um setor de software competitivo, nacional e internacionalmente, é essencial que os empreendedores do setor coloquem a eficiência e a eficácia dos seus processos em foco nas empresas, visando à oferta de produtos de software e serviços correlatos, conforme padrões internacionais de qualidade”, de acordo com o MPS.BR (2011).

Segundo a ISO/IEC 9126, a qualidade deve garantir a funcionalidade, confiabilidade, usabilidade, eficiência, manutenibilidade e portabilidade dos sistemas. De acordo com o CMMI, a garantia da qualidade está diretamente ligada às pessoas, ferramentas e procedimentos. A união e a maturidade dessas dimensões refletem no resultado satisfatório do processo de produção e acompanhamento pós-produção.

A engenharia da qualidade é o conjunto das técnicas e procedimentos para estabelecer critérios e medidas da qualidade de um produto, identificar produtos que não estejam conformes tais critérios, evitando que cheguem ao mercado, e acompanhar o processo de produção, identificando e

eliminando as causas que levaram a não-conformidades. O enfoque tradicional da engenharia da qualidade enfatiza o controle, seja através de inspeções de produto, seja através do controle do processo. Já uma visão mais moderna, preocupa-se com as ações preventivas que possam garantir que a qualidade será alcançada, usando o controle apenas como apoio, quando for indispensável, em um contexto de gestão total da qualidade, estendida a todas as atividades da empresa. (IRWIN, 1965).

As engenharias têm de forma sincronizadas as suas atividades e projetos de construções com outras atividades que são responsáveis pela verificação de produtos intermediários e finais, facilitando a identificação de defeitos que poderão ser removidos. O mesmo acontece com a Engenharia de Software: o desenvolvimento de sistemas computacionais requer atividades de análise e verificação durante todo o ciclo de desenvolvimento, de acordo com Pezzè & Young (2008, p. 25).

Conforme Pressman (1995, p. 786), “A atividade de teste de software é um elemento crítico da garantia de qualidade de software e representa a última revisão de especificação, projeto e codificação”.

A finalidade dos testes para a detecção de defeitos é expor defeitos latentes em um sistema de software antes de o sistema ser entregue. Isso contrasta com os testes de validação, que se destinam a demonstrar que um sistema cumpre com suas especificações. Os testes de validação exigem que o sistema opere corretamente, utilizando os casos de testes de aceitação que foram apresentados. Um teste bem-sucedido para a detecção de defeitos é aquele que faz com que o sistema opere incorretamente e, como consequência, expõe um defeito existente. Isso enfatiza um fato importante sobre os testes e demonstra a presença, e não a ausência, de defeitos de programa. (SOMMERVILLE, 2003, p. 377).

Uma vez que a garantia do processo é planejada e executada, respeitando os modelos, padrões e as normas durante todo o ciclo de vida do projeto, o software desenvolvido também será beneficiado com essa qualidade.

4 APRESENTAÇÃO DOS DADOS

4.1 Produção sem Qualidade

Com a globalização das informações e do consumo, o mercado nos países desenvolvidos e emergentes, visa atender a todas as classes sociais, fazendo com que as empresas passem a buscar formas de atender diversos tipos de usuários com comportamentos diferentes e logo, consumos diferentes. Esse problema tem despertado nas organizações o sentimento de atraso com relação à gestão da qualidade, uma vez que, as metodologias e boas práticas ainda são pouco exploradas e usadas de forma incoerente. A cultura da produção focada em quantidade e a resistência a mudanças são os grandes problemas enfrentados por essas instituições.

Os japoneses aprenderam que a melhor forma de manter um mercado equilibrado não é investir apenas na produção, mas sim, na qualidade de seus processos. Entender que um produto que satisfaz às necessidades dos clientes gera uma imagem positiva no mercado, garantindo a fidelização dos consumidores, foi algo que alavancou os orientais, tornando-lhes referência na indústria mundial.

Mesmo com os problemas que a América Latina enfrentava nos anos 80 — década perdida —, onde a economia mundial teve um crescimento menor do que nos anos 70, várias empresas se tornavam referências no mercado e passavam a investir em novos produtos. As empresas japonesas passaram a ser as grandes precursoras de produtos inovadores. Tal crescimento econômico que norteava o Japão, passou a ser invejado por países europeus e pelos E.U.A, que buscaram formas alternativas de combater o concorrente. Desde então, as empresas passaram a perceber que o grande motivo do alavancar econômico no Japão, nada mais era que a educação voltada à gerência — que os orientais vinham desenvolvendo desde os anos 50. Nos anos 80,

os japoneses além de investirem em novas ferramentas de produção, passaram a investir no capital intelectual, valorizando ainda mais a gestão da qualidade.

No Brasil, o despertar para a gestão da qualidade se daria mais tarde, nos anos 90. Devido à grande demanda de consumo, empresas multinacionais começaram a abrir suas filiais no país, despertando mais emprego. Uma das dificuldades encontradas na região foi a demanda por mão de obra qualificada. Os brasileiros estavam acostumados a produzir para empresas locais, que tinham processos mais rústicos, voltados para a produção em massa. Era necessário despertar nos profissionais brasileiros, o interesse pelos processos industriais. A década citada seria de muitos erros e poucos acertos, mas que serviria de grande experiência para os anos 2000, onde a maturidade dos processos adotados pelas empresas, seria copiada por países vizinhos.

Os anos 2000 não começaram com muitas expectativas, pois o estado intervia cada vez menos no setor industrial, o que deixava os funcionários em nível de desvantagem nos direitos trabalhistas. A partir de 2002, a indústria começava a mudar, devido os grandes investimentos que o governo passava a fazer no setor industrial, agrário e tecnológico, além da intervenção pública nos direitos e deveres dos empregados e empregadores. O país passava por um processo de mudança considerável, adotando filosofias e métodos utilizados em países desenvolvidos. A educação passava a ser priorizada, abrindo novas vagas nas universidades e centros universitários. O poder da moeda transformava o brasileiro em um povo mais consumista, crítico e seletivo.

As empresas de tecnologia, juntamente com as alimentícias, foram as que mais lucraram com o crescimento do país. O governo passou a investir pesado em novos negócios e abriu isenção de impostos para as empresas que destinassem verbas para projetos de desenvolvimento de sistemas ou robótica no

país.

Várias empresas de tecnologia fechavam em menos de um ano, devido a fatores como: projetos com custos elevados, produtos que não atendiam mais às necessidades dos clientes, requisitos mal-interpretados, produtos com porcentagem significativa de defeitos. Em contraponto, outras empresas nasciam, tornando o mercado competitivo e mais desafiador. A partir daí, as empresas de tecnologia do país passavam a observar a gestão da qualidade voltada para software como uma solução para os problemas. Uma forma de "gerenciar" que estava dando certo no setor alimentício e que certamente daria certo no setor da tecnologia.

4.2 Mudanças na Forma de Produzir

O desenvolvimento de software sempre foi muito voltado para a construção em si, não valorizando a importância da gestão do projeto. Mas um bom projeto de tecnologia nasce de um bom planejamento, onde representantes de todas as áreas envolvidas no desenvolvimento — cliente, gerente do projeto, líder técnico, líder de testes, analista da qualidade — devem estar inseridos nas etapas iniciais — negociação, modelagem de negócio, design do negócio, levantamento de requisitos e etc.

Com a concorrência em alta, é comum uma empresa ficar com uma fama ruim e não conseguir mais se reerguer. Na era da internet, um comentário ruim sobre uma marca ou um produto, pode ter sérios danos financeiros tanto para a empresa que disponibilizou o software, quanto para a empresa que desenvolveu. Pensando nisso, as empresas de tecnologia vêm buscando formas de melhorar esse processo de construção, com a contribuição dos stakeholders — todos os envolvidos no projeto.

No início, a implantação da área da qualidade passou por diversos desafios, o maior deles foi sem dúvida o financeiro. As empresas viam a área como um fator necessário em seus processos, mas que requeria um alto

investimento para benefícios em longo prazo. Um outro fator que dificultava a implantação da área era a resistência a mudanças por parte dos profissionais, mesmo sentindo a necessidade de mais organização e segurança das atividades que estavam sendo executadas. Com o tempo a área foi ganhando espaço dentro das organizações, sendo hoje, disciplina da grade curricular dos cursos de informática nas grandes universidades do país.

4.3 Gestão da Qualidade

A gestão da qualidade de software tem como objetivo normatizar e rever periodicamente os conceitos e métodos utilizados por uma empresa, garantindo melhoria e correções nos processos de desenvolvimento, com base em estudos e estatísticas minuciosas. A área acompanha todos os processos de construção dos softwares: modelagem de negócio, elicitação de requisitos, análise e designer do projeto, implementação, testes, implantação, gerenciamento de configuração e mudança, gerenciamento de projeto e ambiente.

A área da qualidade trabalha de acordo com as normas da ISO 9001, ISO 9002 e ISO 9003. Apesar de muitos acharem que os procedimentos de certificação de uma empresa de tecnologia é muito diferente das empresas industriais, os profissionais vêm mostrando que não.

As fábricas de softwares são os grandes exemplos disso, pois as mesmas têm se preocupado com a focalização nos clientes, abordagem dos processos, envolvimento dos stakeholders, comprometimento da liderança, relação com os fornecedores e a melhoria contínua dos processos.

Muitos clientes que prezam por mais confiabilidade nos contratados, buscam empresas que sejam certificadas na área. Hoje são diversas as certificações, onde cada uma atende necessidades específicas do negócio.

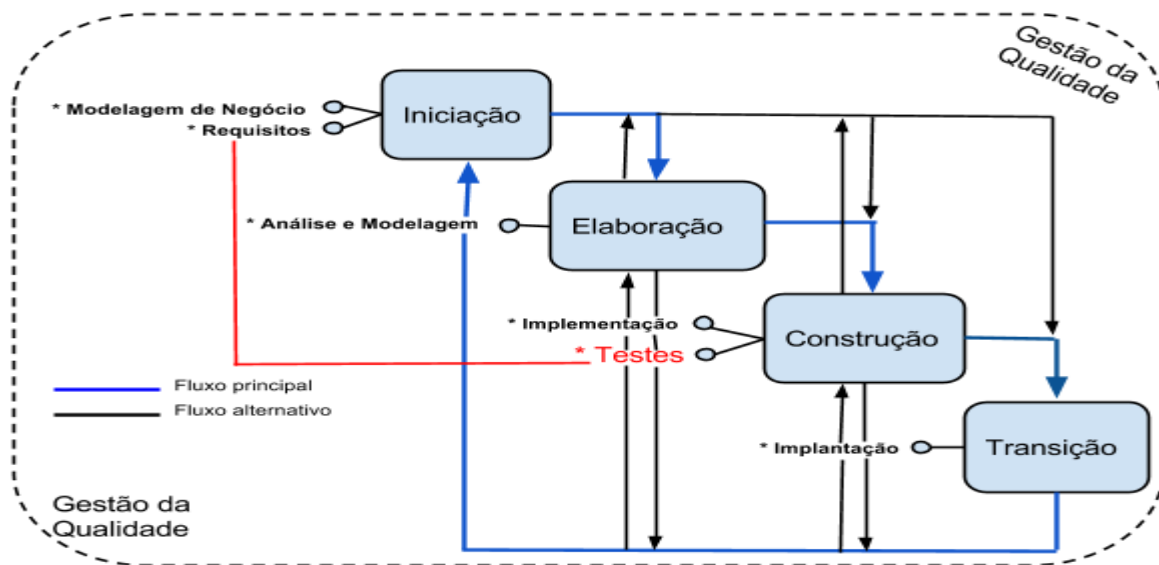
As mais conhecidas da área de tecnologia são: CMMI e MPS.BR. Para se certificar, a empresa precisa atender algumas exigências e demonstrar maturidade nos processos de produção. Essas certificações além de darem um peso maior ao currículo da empresa, atraem clientes que buscam por parceiros com maior qualidade nos processos.

A área da qualidade se responsabiliza pela garantia do cumprimento dos processos e pela melhoria dos mesmos — melhoria contínua. O produto é visto como consequência

da execução desses processos. Apesar dos esforços serem voltados para o processo, o principal objetivo é garantir um produto final que satisfaça o cliente.

A garantia da satisfação do cliente é de responsabilidade da gestão da qualidade, de forma indireta, e da engenharia de testes, de forma direta, conforme é possível ver na Figura 1. Dentro do processo de desenvolvimento, a garantia da qualidade precisa gerenciar as subáreas, garantindo que as mesmas cumpram suas responsabilidades com eficiência.

Figura 1 — Exemplo de um processo de desenvolvimento de software



4. 4 Engenharia de Testes

A área de testes, também conhecida como engenharia de testes, tem como objetivo prover informações sobre a qualidade do produto implementado, garantindo que o mesmo atenda aos requisitos especificados. Para chegar a tal conclusão, métodos são criados para uma profunda e minuciosa análise das partes e do produto como um todo. É necessário que a engenharia de testes esteja integrada e inserida em todas as partes que envolva a produção do projeto, desde o levantamento dos requisitos até a transição final.

A área passou a ser implantada pelas

fábricas de softwares quando nos anos 90 muitas empresas começaram a receber reclamações de seus softwares, pois apresentavam muitos defeitos. Muitas vezes, a qualidade do processo de produção era boa, gerava um produto bom, mas que não atendia aos requisitos do cliente. Ainda hoje é possível encontrar empresas que não possuem profissionais da área, chegando a entregar seus produtos sem uma análise que garanta a qualidade dos mesmos.

O produto não pode ser avaliado apenas por seus fatores externos, mas também por seus fatores internos. Sendo assim, a análise pode e deve ser feita separada por módulos, onde cada

profissional é especializado em um tipo de teste ou em vários tipos. Baseado nas normas da ISO 9126, alguns atributos são analisados por essa área:

- **Funcionalidade:** o sistema deve conseguir executar as funcionalidades para o qual foi criado. Ex.: uma calculadora não é obrigada a calcular a média aritmética, mas tem por obrigação: somar, dividir, multiplicar e subtrair;
- **Confiabilidade:** o sistema deve manter um desempenho que seja aceitável — o desempenho “aceitável” deve ser definido no planejamento —, garantindo a maturidade do software;
- **Usabilidade:** além de ser atraente para o usuário, o produto deve ser intuitivo;
- **Eficiência:** é necessário avaliar o comportamento do sistema com relação ao tempo de execução;
- **Manutenibilidade:** o sistema deve dar suporte a ajustes, melhorias e atualizações;

- **Portabilidade:** deve ser avaliada a capacidade de adaptação do produto ao ser transferido para outro ambiente.

O objetivo da engenharia de testes não é confirmar que o software está em perfeito estado, uma vez que, testes não garantem a perfeição, mas sim, a qualidade do que está sendo desenvolvido — isso porque alguns erros só irão aparecer quando o produto já estiver no mercado. A realização de testes exaustivos também é quase impossível — testes exaustivos são testes que cobrem todas as combinações de entradas e condições —, pois será necessário muito tempo de execução e muitos recursos, sendo difícil encontrar um projeto que disponibilize muitos ativos para a execução dos testes. Isso não quer dizer que o software deva conter muitos erros. Existe um planejamento que determina os tipos de erros que não podem ser aceitos no produto e uma margem de erros que o mesmo pode conter. O Quadro 1 mostra alguns tipos de erros aceitáveis e outros não aceitáveis.

Quadro 1 — Exemplos de erros não aceitáveis e aceitáveis

Erros não aceitáveis (na maioria das vezes):	Erros aceitáveis (na maioria das vezes):
<p>A. Layout quebrado;</p> <p>B. Sistema com problemas na segurança;</p> <p>C. Requisitos não implementados;</p> <p>D. Persistência dos dados sem sucesso;</p> <p>E. Performance muito baixa.</p>	<p>A. Layout quebrado de web site, quando utilizado em um navegador específico;</p> <p>B. Problemas no sistema quando utilizado em um sistema operacional específico;</p> <p>C. Lentidão do sistema quando executado com um outro programa rodando simultaneamente.</p>

4.5 Hierarquia, Cargos e Responsabilidades

Os profissionais de testes não têm por função garantir que o software não possui erros, mas sim o contrário. Esses têm por função encontrar erros, mesmo sabendo que quanto mais erros encontrados, mais trabalho será gerado para a equipe de desenvolvimento. Algumas empresas chegam a premiar

funcionários que conseguem bater metas de quantidade de inconsistências encontradas.

O trabalho de testes não começa a partir do momento em que o produto é entregue para ser testado, mas logo após a definição dos requisitos, passando por todo um processo de planejamento.

Os gerentes de testes — também conhecidos como líderes de testes — têm como

objetivo realizar todo o planejamento dos testes, levantando qual será a estratégia de testes, os tipos de testes utilizados, quais serão as tecnologias adotadas pela equipe para a execução das atividades, a definição do ambiente em que os testes serão executados, o cronograma e as prioridades.

Os analistas de testes têm por função escrever os cenários ou casos de testes, deixando claro todos os passos que devem ser realizados na execução dos testes.

Os arquitetos de testes devem garantir que o ambiente de execução esteja de acordo com o especificado no plano de testes, disponibilizando recursos de hardware e software, removendo impedimentos, de forma a garantir que os testadores possam executar suas atividades.

Os automatizadores de testes são responsáveis por criar scripts automatizados, que serão executados posteriormente.

Os testadores — conhecidos como engenheiros de testes ou técnicos em testes — têm por função executar os testes, de forma a encontrar erros, reportá-los para os responsáveis, retestá-los, despertar melhorias e dar um feedback à equipe sobre a qualidade do

produto, baseado na quantidade de inconsistências encontradas e resolvidas.

Hoje no mercado existem empresas que prestam serviços de testes, trabalhando em parceria com empresas que só desenvolvem software, cabendo às primeiras a responsabilidade de encontrar inconsistências no que foi desenvolvido.

Essas empresas ainda são raras e muito pouco procuradas, porque geralmente a própria empresa desenvolvedora contém equipes de testes.

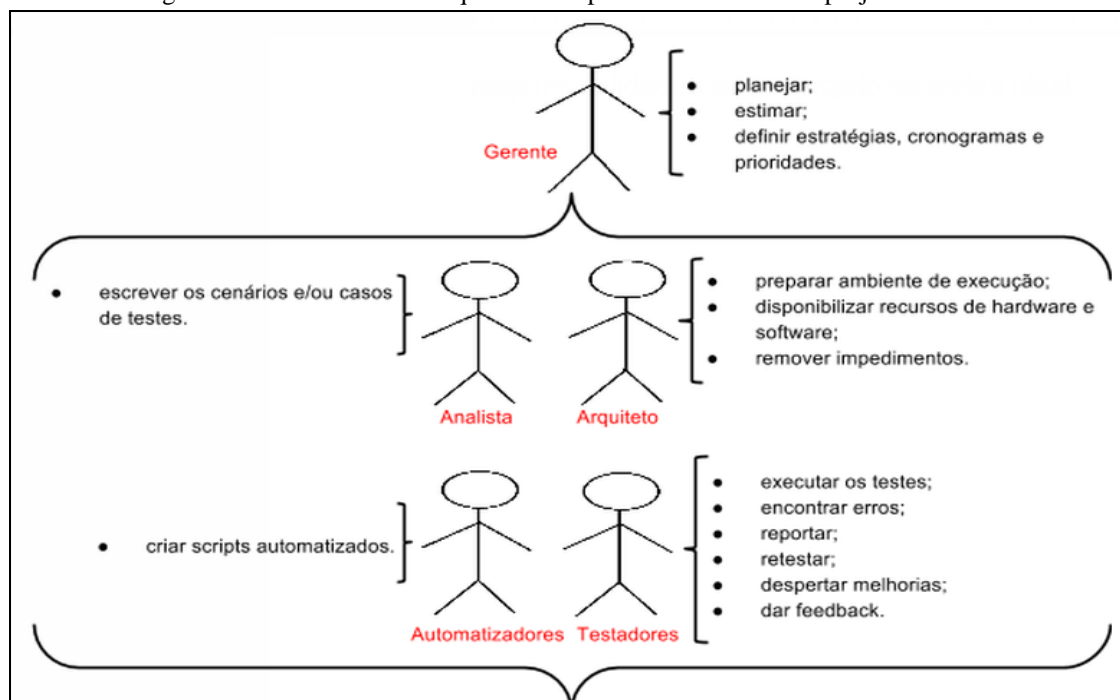
Já é possível encontrar nessas empresas, projetos apenas de testes, onde os testadores não interagem de forma direta com os desenvolvedores.

Nesses dois casos citados, é possível encontrar equipes com vários tipos de profissionais da área, podendo essas equipes serem denominadas “independentes”.

Mas o comum é encontrar profissionais que exerçam mais de uma função na área — os analistas de testes geralmente fazem as atividades dos engenheiros e dos arquitetos.

A Figura 2 mostra um modelo de hierarquia e de responsabilidades de um projeto de testes ideal.

Figura 2 — Modelo de hierarquia e de responsabilidades de um projeto de testes



O tipo de projeto apontado na figura 2 não é muito comum, pois requer um investimento muito alto. Geralmente esse tipo de projeto é adotado para produtos críticos, que põem em risco vidas ou o financeiro do cliente.

4.6 Planejamento

Os testes não se limitam apenas à execução, os mesmos exigem todo um planejamento, que envolve profissionais disponíveis para a criação das documentações. Quanto mais complexo um teste, mais complexo será a sua documentação, por esse motivo, é importante que os responsáveis pelos testes estejam envolvidos na elicitação dos requisitos. Todo o fluxo dos testes deve estar especificado no plano de testes. Esse plano deve seguir o padrão IEEE 829, que é responsável pelo Padrão de Documentação de Teste de Software. O mesmo deve conter os seguintes itens:

- Identificador do documento;
- Introdução;
- Funcionalidades que serão testadas;
- Funcionalidades que não serão testadas;
- Estratégia de testes;
- Ferramentas utilizadas nos testes;

- Critérios de sucesso, falha, suspensão;
- Produtos dos testes;
- Artefatos gerados no processo de testes;
- Tarefas de testes;
- Ambiente de testes;
- Papéis e responsabilidades;
- Cronograma;
- Riscos;
- Contingências;
- Aprovações.

Na estratégia de testes é possível definir os tipos que serão adotados. Alguns testes onde os sistemas atuam em projetos críticos, necessitam de um processo mais minucioso, muitas vezes, havendo a necessidade do testador simular o ambiente do cliente. Esses testes requerem muito tempo para serem executados e necessitam de uma infraestrutura avançada. Existem os testes automatizados, onde os engenheiros precisam entender de programação para que possam codificar algo no produto desenvolvido.

São inúmeros os tipos de testes e cada caso aborda uma necessidade específica. O Quadro 2 aponta alguns tipos de testes e seus respectivos objetivos.

Quadro 2 — Tipos de testes

Tipo de Teste:	Descrição:
Teste de Unidade	Tem como objetivo validar componentes ou classes. Foca em uma parte do código.
Teste de Integração	Tem como objetivo validar a junção das partes analisadas no teste de unidade, ou seja, testar a integração das unidades.
Teste Operacional	Tem como objetivo analisar a qualidade do software depois de horas ou dias de utilização.
Teste Positivo-negativo	Tem como objetivo garantir que o sistema vai executar o “caminho feliz” — caminho de sucesso da operação — e seus fluxos de exceção.
Teste de Regressão	Tem como objetivo re-executar os testes. É realizado após a execução de todos os testes, para garantir que durante o desenvolvimento, erros

	novos não surgiram ou erros antigos não voltaram a acontecer.
Teste Caixa-branca	Tem como objetivo analisar o código da aplicação, de forma a garantir que padrões de codificação estejam sendo seguidos e que nenhuma brecha de segurança está presente.
Teste Caixa-preta	Foca na funcionalidade em si, não se importando com o código-fonte. Preocupa-se com o resultado apresentado na aplicação, comparando o mesmo com o resultado esperado.
Teste Funcional	Focado nas funcionalidades, mas de uma forma mais ampla, tendo como base o documento de requisitos. Serve para validar se o requisito implementado condiz com o que foi solicitado pelo cliente.
Teste de Interface	Tem como foco analisar a usabilidade da interface implementada, validando componentes que fazem parte dessa interface.
Teste de Performance	Tem como objetivo garantir que o tempo de resposta da aplicação é aceitável.
Teste de Carga	Analisa o sistema, quando vários usuários estão utilizando o mesmo de forma simultânea.
Teste de Aceitação	Realizado no ambiente do cliente por futuros usuários do sistema, de forma a ter um retorno do cliente sobre o que foi desenvolvido.
Teste de Volume	Tem por objetivo, analisar a quantidade de dados suportada pelo sistema, assim como a capacidade de armazenamento do mesmo.
Teste de Estresse	Realizado em situações esporádicas, cujo objetivo é conhecer os limites da aplicação.
Teste de Configuração	Analisa o comportamento do sistema em várias plataformas.
Teste de Instalação	Avalia a instalação do sistema.
Teste de Segurança	Garante a confiabilidade do sistema. Técnicas são utilizadas para encontrar brechas de segurança na aplicação.

4.7 Escrita dos Casos ou Cenários de Testes

Quando todo o plano já estiver concluído, cabe aos analistas criarem os casos de testes — conjunto de condições, entradas e saídas dos testes —, que servirão de guia para o trabalho dos engenheiros. Algumas empresas que adotam metodologias ágeis vêm substituindo os casos de testes por cenários de

testes, por serem mais rápidos, mais objetivos e sem muitas regras de escrita. Mas também existem empresas que utilizam uma mistura dos dois artefatos, sendo os cenários, documentos que servem de base para a escrita dos casos de testes. A Figura 3 mostra um exemplo de caso de teste e a Figura 4 mostra um exemplo de cenário de teste.

Figura 3 — Exemplo de caso de teste

CASO DE TESTE	
ID	TC001
Descrição	Cadastrar Usuário
Ambiente	Navegador Chrome Versão 30.0.1599.101 m
Ator	Administrador do Sistema
Precondição	Ter menos de 10000 usuários cadastrados.
Procedimentos	1 - pressione o botão [Cadastrar Usuário]; 2 - preencha o campo [Nome]; 3 - preencha o campo [Usuário]; 4 - preencha o campo [Senha]; 5 - preencha o campo [Confirma Senha]; 6 - Selecione um tipo no combo-box [Tipo de Usuário]; 7 - Pressione o botão [Cadastrar].
Pós-condições	Usuário cadastrado com sucesso.

Figura 4 — Exemplo de cenário de teste

CENÁRIOS DE TESTES	
ID	TC001
Descrição	Cadastrar Usuário com Sucesso
ID	TC002
Descrição	Cadastrar Usuário com Falha no Banco
ID	TC003
Descrição	Cadastrar Usuário com Senhas Incorretas

Resumidamente, a diferença entre um caso e um cenário de teste, é que o primeiro descreve “como” uma parte do produto deve ser testada, e o segundo descreve “o que” deve ser testado. Os dois tipos de artefatos citados não são obrigatórios, pois existem inúmeros documentos que servem de guia para os engenheiros.

4.8 Preparação do Ambiente de Testes

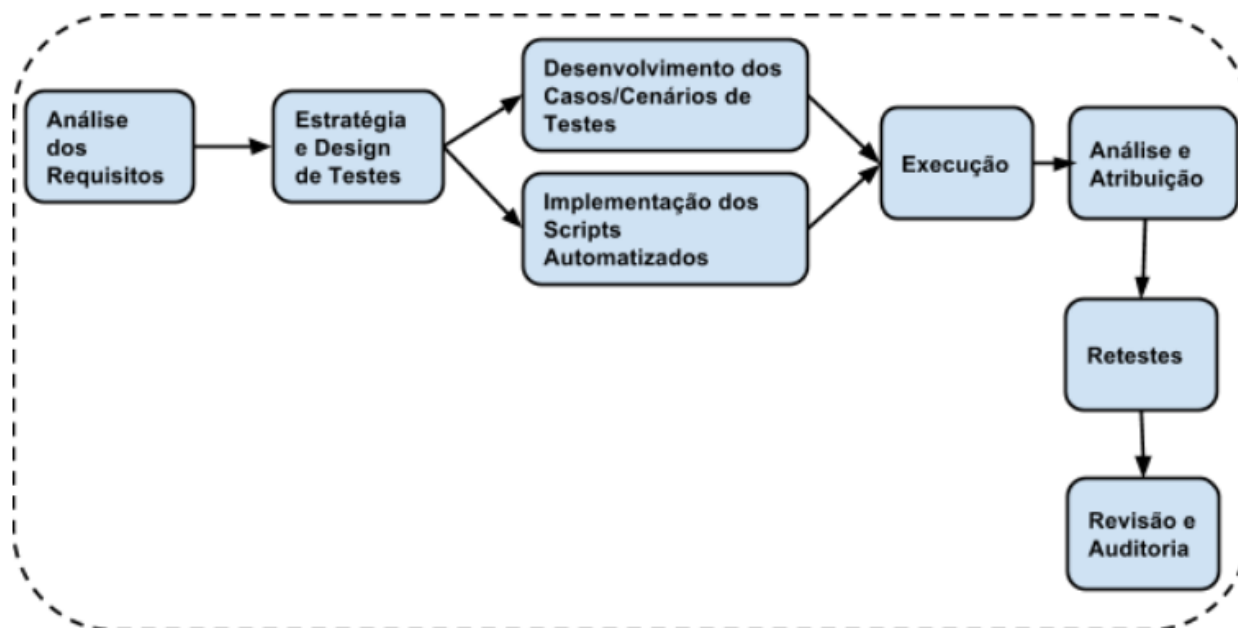
Após a definição do planejamento e da escrita dos casos ou cenários, é preciso que o arquiteto disponibilize o ambiente para a execução dos testes. Muitas vezes, essa tarefa é um pouco complicada, pois requer que o arquiteto simule um ambiente muito próximo do

ambiente de produção, necessitando de recursos como hardware e periféricos com custos elevados e de difícil compreensão.

4.9 Execução e Reportagem

Quando todo o ambiente estiver disponível, os engenheiros devem executar os casos de testes, utilizando as ferramentas disponíveis para isso. Ao encontrar uma inconsistência, é necessário reportá-la aos desenvolvedores, para que esses possam implementar as devidas correções. Cada empresa adota uma forma de tratar as inconsistências encontradas, não existindo um fluxo padrão. A Figura 5 mostra um tipo de fluxo muito utilizado nas fábricas de software.

Figura 5 — Modelo de ciclo dos testes



As melhorias também devem ser despertadas, uma vez que, a equipe de testes está inserida na hora da especificação e muitas vezes, acaba conhecendo o sistema melhor do que os desenvolvedores. Algumas melhorias e inconsistências não podem ser implementadas sem a permissão do líder técnico, já que a mudança pode causar um grande impacto no software — cada empresa tem uma forma de trabalhar com esse fluxo de mudança. Algumas ferramentas podem ser utilizadas para auxiliar nesse gerenciamento de reportagem de erros e melhorias.

É possível incluir a engenharia de testes em métodos de produção já definidos e amplamente utilizados por indústrias, como o PDCA, 6 Sigmas e o Método DMAIC.

4.10 Utilizando o Ciclo PDCA na Modelagem e Análise de Testes

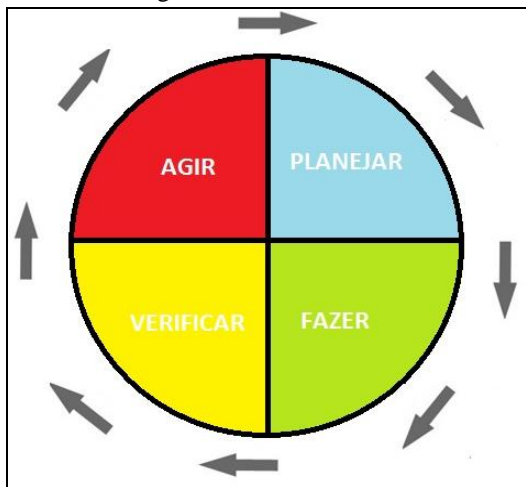
O ciclo PDCA consiste em uma sequência de passos a serem seguidos, permitindo controlar um processo já definido, conforme é possível observar na Figura 6. A modelagem e análise de testes seguem um fluxo parecido com o do PDCA, podendo até ser

adotado para a criação do planejamento e para a execução dos testes. O ciclo se divide em:

- A. **Planejar:** nesta fase, a equipe planeja os critérios que serão abordados nos testes, os resultados almejados e quais as perguntas que devem ser respondidas ao final do processo. Os critérios de saídas são apurados pelos históricos de testes passados e pela experiência da equipe de testes envolvida;
- B. **Fazer:** nesta fase, o planejamento começa a sair do papel, ou seja, começa a execução dos testes. Os estudos e análises feitos na primeira fase começam a ser aplicados, mobilizando a equipe de testes que fará as prescrições, coletando todos os resultados obtidos, para uma futura análise a ser usada nas próximas fases;
- C. **Verificar:** nesta etapa se verifica o que se aprendeu durante a execução do plano de teste. Também é nessa fase que se compara as previsões feitas na etapa de planejamento, verificando se os resultados obtidos foram os resultados esperados;
- D. **Agir:** nesta fase é feita uma análise dos resultados obtidos, para a implementação de um plano de ação para as não conformidades

encontradas. A equipe de testes com seu líder decide o que deve ser reportado com urgência e se a release será liberada ou não.

Figura 6 — Ciclo PDCA



4.11 Six Sigma aplicado na Engenharia de Testes

O Six Sigma foi uma criação da Motorola com o intuito de quebrar um paradigma: tentar que todas as áreas da empresa conquistassem a meta de zero defeitos nos seus processos. O objetivo do Six Sigma é ter uma melhor eficiência nos seus processos, provendo uma alta qualidade nos produtos e serviços a um custo reduzido, tentando amenizar a variabilidade dos processos.

Essa metodologia tem o foco na satisfação do cliente, mantendo a qualidade e reduzindo o ciclo do processo que demorar mais tempo para ser entregue ao cliente e com isso atrasando ou prejudicando a produtividade da empresa.

O segredo do Six Sigma é que todos são envolvidos na aplicação do programa, principalmente a alta administração — os diretores e os líderes do processo. Todos os envolvidos recebem um papel no processo, esses papéis são:

A. **Champions:** são gerentes das diferentes áreas da empresa, que definem

um plano de mudanças, melhorias ou soluções a serem aplicadas;

B. **Master Black Belts:** são profissionais que auxiliam os champions, dedicando um tempo integral no suporte às equipes e aos líderes de equipe (Team Leader);

C. **Black Belts:** são responsáveis por pôr em prática a metodologia, com o objetivo de alcançar o Six Sigma nos seus projetos. Os black belts dividem a responsabilidade com os membros da sua equipe, incentivando todos a quererem alcançar o nível de excelência Six Sigma;

D. **Green Belts:** são liderados pelos black belts. Eles participam de projetos menos complexos, auxiliados a uma ou mais equipes de acordo com a experiência e conhecimento. Possuem responsabilidades como membros dos projetos.

4.12 Método DMAIC

O DMAIC é um método de análise que tem como seu principal papel agir em ações corretivas, ou seja, tomar ações para os problemas que forem encontrados ao logo do processo DMAIC. O método tem como objetivo a retroalimentação do ciclo, no qual mensuraria os defeitos e falhas encontrados para análise e em seguida a correção. Esses defeitos e falhas devem ser controlados, para que nos novos ciclos os mesmos não se repitam. Esse método se divide em 5 fases ou etapas:

A. **Definir:** esta fase é voltada para a definição do problema a ser resolvido ou solucionado, onde também é avaliado o impacto econômico que a falha irá acarretar ao projeto;

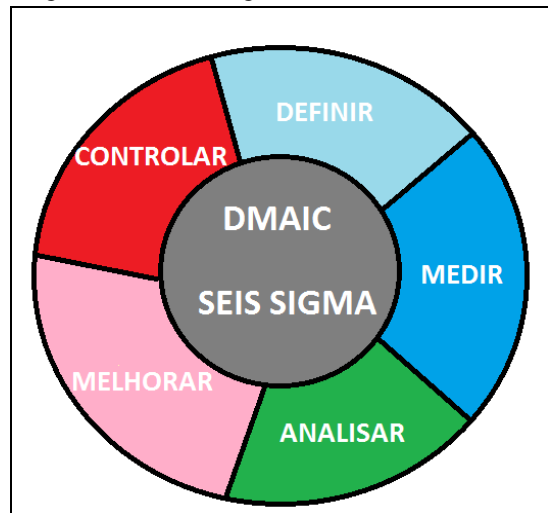
B. **Medir:** esta etapa tem como principal objetivo, localizar a raiz do problema, ou seja, em qual módulo o problema (falha) está ocorrendo;

C. **Analisar:** nesta fase, o objetivo principal é identificar as causas geradoras do problema e quais melhorias devem ser implementadas, utilizando os dados coletados das etapas anteriores. Também podem ser usadas ferramentas matemáticas e de estatísticas para a coleta dos dados;

D. **Melhorar:** esta etapa tem como objetivo, implementar as soluções propostas para o problema analisado, no qual serão aplicadas as melhores ideias para solucionar as inconsistências encontradas na etapa de análise. As soluções escolhidas pela equipe são testadas em um outro ambiente com uma pequena amostragem de dados para validação, e a partir dos resultados obtidos, são aplicados os ajustes e as melhorias. Após a aplicação das soluções, se a mesma continuar não favorável, o problema volta para a fase de medição;

E. **Controlar:** nesta etapa, são monitoradas as implementações das soluções do problema para garantir que o sistema continue funcionando com sucesso, sem falhas. Caso na etapa de controle se encontre alguma outra inconsistência, deve-se voltar para a etapa de medição, realizando assim, uma nova investigação do problema. Caso a solução continue funcionando com sucesso e os problemas corrigidos, o projeto segue para uma nova implementação, que é a padronização das alterações realizadas para conserto do problema em outros módulos ou em parte do projeto, utilizando o método DMAIC de forma corretiva com uma retroalimentação, onde em cada ciclo o processo irá melhorar sempre de forma contínua. A Figura 7 mostra como essas fases estão integradas.

Figura 7 — Modelo gráfico do Método DMAIC



5 CONCLUSÃO

Testes ainda é uma área nova e por isso, pouco explorada. Os programas, metodologias e ferramentas adotadas, ainda são escolhidas baseadas em experiências de profissionais autodidatas. Apesar de já existir alguns livros na área, ainda são poucos quando comparados à quantidade de material disponível nas subáreas de TI, como engenharia de sistemas e design gráfico. Ainda é preciso mais investimentos no setor e impulso dos órgãos públicos. Mesmo já provendo de instituições públicas exclusivas para o desenvolvimento de sistemas, o governo ainda é um órgão atrasado na área da computação, onde o progresso limita-se ao mercado competitivo das empresas desenvolvedoras de software, onde dessas, apenas as que realmente se preocupam com a qualidade dos produtos desenvolvidos, contam com uma área exclusiva para desenvolvimento de testes.

Ainda é preciso entender que a área pode ser usada para garantir a qualidade do produto, não só por um fator econômico, mas também pela reputação da empresa. A importância da engenharia de testes para a garantia da qualidade é uma fator pouco vislumbrado, uma vez que, poucas empresas têm a visão da

qualidade de software. Mas olhando o histórico das empresas que adotaram a área, utilizando políticas institucionais bem definidas em um ambiente que respeita os processos, é possível ver que os benefícios são inúmeros. Alguns estudiosos chegam a atribuir os grandes problemas dos sistemas governamentais à ausência da área de testes. O que de fato é uma grande verdade, pois muitos dos erros chegam a ser considerados básicos — erros que não passariam despercebidos por uma equipe de testes.

A adoção da área requer um investimento que trará lucros em longo prazo, uma vez que a mesma garantirá a qualidade dos produtos desenvolvidos e a manutenção ou o crescimento do valor da marca. Existe também, a dificuldade em encontrar mão de obra especializada, que saiba trabalhar com as diversas ferramentas disponíveis, dentro de diferentes contextos de projetos.

Uma boa política de produção na engenharia de software requer investimentos na área da qualidade, e uma boa gestão da qualidade voltada para software, requer uma boa visão estratégica da implementação da engenharia de testes.

6 REFERÊNCIAS

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **NBR ISO 9000**: Sistemas de gestão da qualidade - fundamentos e vocabulário. Rio de Janeiro, 2000.

Ciclo PDCA. In: **CEDET**. Campinas: 2014. Disponível em: <<http://www.cedet.com.br/index.php?/Tutoriais/Gestao-da-Qualidade/ciclo-pdca-plan-do-check-act-planejar-fazer-verificar-agir.html>>. Acesso em: 11 jan. 2014.

CMMI. In: **CMMI Institute and Clearmodel**.

Pensilvânia: 2014. Disponível em: <<http://certification.cmmiinstitute.com/>>. Acesso em: 10 jan. 2014.

COBIT. In: **ISACA**. Estados Unidos da América: 2014. Disponível em: <<http://www.isaca.org/COBIT/Pages/default.aspx>>. Acesso em: 10 jan. 2014.

DMAIC - Ação corretiva (Método de análise e solução de problemas - MASP). In: **Administradores**: 2010. Disponível em: <<http://www.administradores.com.br/mobile/artigos/administracao-e-negocios/dmaic-acao-corretiva-metodo-de-analise-e-solucao-de-problemas-masp/46167/>>. Acesso em: 11 jan. 2014.

IRWIN, R. D. Quality Control and Industrial Statistics. **Library of Congress**. Homewood, Illinois. 1965.

ISO/IEC 9126. In: **ISO.org**. Suíça: 2014. Disponível em: <<http://www.iso.org/iso/>>. Acesso em: 10 jan. 2014.

MPS.BR. In: **Softex**. Brasília: 2013. Disponível em: <http://www.softex.br/wp-content/uploads/2013/07/MPS.BR_Guia_de_Implementacao_Parte_8_20111.pdf>. Acesso em: 8 dez. 2013.

PERSPECTIVAS: o futuro da gestão da qualidade. **Falando de qualidade**, São Paulo, v. 14, n. 151. 2004.

PEZZÈ, Mauro; YOUNG, Michal. **Teste e Análise de Software**: processos, princípios e técnicas. Porto Alegre: Bookman, 2008. p.25.

PRESSMAN, Roger S. **Engenharia de Software**. São Paulo: Pearson Makron Books, 1995. p.786.

PRESSMAN, Roger S. **Engenharia de Software**. 6. ed. Porto Alegre: AMGH, 2010.

SOMMERVILLE, Ian. **Engenharia de Software**. São Paulo: Addison Wesley, 2003. p.377.

Artigo submetido em: 10.02.2014

Artigo aprovado para publicação em: 03.09.2015