10.5748/17CONTECSI/PSE-6592

CORRELATION OF SOFTWARE TESTS WITH REDUCTION OF POST-SOFTWARE IMPLEMENTATION INCIDENTS - A CORRELAÇÃO DOS TESTES DE SOFTWARES COM A REDUÇÃO DE INCIDENTES PÓS-IMPLANTAÇÃO DE SOFTWARE

Victor Oliveira Cipriano - Centro Universitário Estácio Do Ceará - victorcuru@gmail.com
Wellington Sousa Aguiar - Centro Universitário Estácio Do Ceará - wellington@tecsist.com
José MãRio Bezerril Fontenelle - Centro Universitário Estácio Do Ceará - jomabefon@gmail.com
Péricles Alves Ferreira De Arruda - Centro Universitário Estácio Do Ceará - pericles@gmail.com
Henrique Nogueira Da Gamma Mota - Centro Universitário Estácio Do Ceará - mota.henrique@gmail.com

INTRODUÇÃO

Partindo do princípio de melhoria das condições e rotinas no âmbito organizacional e humano, empresas de tecnologia investem massivamente em recursos estratégicos e técnicos no intuito de criar respostas prontas e imediatas para as mais diversas necessidades do mercado, a fim de otimizar as experiências dos usuários com os produtos e serviços de TI.

Ao longo do tempo, empresas de desenvolvimento enfrentam desafios para encontrar a qualidade ideal dos produtos de software. A busca pela melhoria contínua dos processos de gerenciamento, dentre outros fatores, elevaram a capacidade das empresas de saírem do caos e da desordem, elevando seu nível de maturidade (SANDHOF, 2004).

Pressman (2011) afirma que as definições relativas aos prejuízos que um software possa vir a impactar devido a sua má qualidade variam, alguns poucos defeitos por menores que sejam, são suficientes para que um programa seja executado de forma inadequada. A ocorrência de erros encontrados a cada 10 linhas de código escrito, multiplicados por milhões de linhas de código em vários produtos comerciais, deduz-se que o custo dos fornecedores de software será de pelo menos a metade dos seus orçamentos para a realização dos testes e correção dos erros.

Devido à alta complexidade dos processos de software, a importância e a necessidade de compor pessoas é uma obrigação útil, para tomar decisões e fazer julgamentos, além de não existir um processo ideal, a maioria das organizações desenvolvem seus próprios processos de desenvolvimento de software. Os processos evoluem de maneira a tirar melhor proveito da capacidade das pessoas em uma organização, assim como das características específicas dos sistemas em desenvolvimento. (SOMERVILLE, 2011).

"Em qualquer processo de desenvolvimento de software, a atividade de testes é fundamental para garantir o sucesso do trabalho. Trata-se de uma atividade bastante custosa, que pode consumir grande parte do esforço para a produção de um sistema. Entretanto a descoberta tardia de defeitos em softwares nos ambientes de produção tem um impacto muito maior, causando mais prejuízos e maior esforço posterior para a correção dos defeitos" (FILHO, 2007, p. 28).

Os Testes são imprescindíveis na criação de um sistema, basicamente possuem a finalidade de avaliar a qualidade do software de acordo com aquilo que foi especificado no projeto, buscando contornar eventuais falhas que possam ocorrer. Uma equipe de testes de software é composta por gerentes, analistas e testadores, que são responsáveis por planejar, supervisionar, registrar e analisar todas as etapas de testes no intuito de gerar novas descobertas

de erros reais que podem ocorrer em ambientes de produção. São elaborados planos de testes com o propósito de colaborar com o desenvolvimento do software, é por meio dele, que todos os componentes técnicos, funcionais e estruturais serão verificados de modo a garantir o funcionamento do programa.

Segundo Filho (2007), para buscar maior aceitação e confiança na forma esperada do comportamento do software, um processo de testes inclui etapas de projeto, execução e análise dos resultados obtidos. Embora muito importante, os testes ainda não possuem ferramentas eficazes que garanta que um programa esteja completamente correto.

"O uso eficaz da TI e a integração entre sua estratégia e a estratégia do negócio vão além da ideia de ferramenta de produtividade, sendo na maioria das vezes fator crítico de sucesso. Hoje, o caminho para este sucesso não está mais relacionado somente com o hardware e o software utilizados, ou ainda com metodologias de Desenvolvimento, mas com o alinhamento da TI com a estratégia e as características da empresa e de sua estrutura organizacional" (PIRES VASCONCELOS; HIKAGE *et. al.*, 2003, p. 30).

Moraes (2010) acredita que a necessidade de implementar Governança de TI dentro das organizações foi um fator motivacional a partir do acelerado desenvolvimento caracterizado pela competitividade, concorrência, exigência dos consumidores, redução de tempo de vida útil dos produtos e uma maior transparência nos processos de negócio.

"A definição de um processo de GTI pode utilizar várias metodologias, tais como COBIT, ITIL, CMM e outras, de modo que se complementem e amenizem suas limitações. Para Mingay e Bittinger (2002), os modelos podem ser combinados para fornecer uma estrutura mais consistente de GTI, estrutura de melhores práticas e controle para a administração de TI" (MORAES, 2010, p.52).

Seguindo a utilização das metodologias utilizadas pela GTI, abordar-se-á no referencial teórico desta pesquisa, a importância e suas respectivas contribuições dentro do ambiente organizacional. Mas especificamente, será extraído da biblioteca ITIL o gerenciamento de incidentes para composição de algumas análises realizadas neste estudo. Os incidentes são basicamente qualquer eventualidade inesperada que possa ocorrer em algum serviço, seja uma interrupção, queda da qualidade ou indisponibilidade da internet, por exemplo. Reconhecer e saber identificar um incidente é importante para que as atividades daquele serviço retornem o mais breve possível.

Aplicar esse conjunto de metodologias e boas práticas e integrá-los aos processos de desenvolvimento de software dentre outras demandas da TI, é importante para adquirir ganho de performance entre as equipes e atividades relacionadas as entregas de curto prazo, sem perca de valor na qualidade do produto ou serviço.

Nesta perspectiva, diante dos desafios enfrentados por empresas que desenvolvem softwares estando restritas a uma estrutura organizacional baseada nas metodologias e boas práticas de gestão de TI, percebe-se a necessidade de avaliar os efeitos resultantes dos testes na qualidade do software após implantação.

Esta pesquisa tem por objetivo levantar e analisar a importância da equipe de testes na implantação de um software. Para tanto foram delineados os seguintes objetivos específicos: adotar a prática da pesquisa científica como motivação para elaboração de novos debates acerca do tema proposto; trazer para a academia a experiência de empresas com relação as práticas adotadas no processo de desenvolvimento e qualidade de software; compartilhar a experiência de profissionais nos seguimentos da TI.

Através de uma abordagem qualitativa e estudos bibliográficos, esta pesquisa tem como propósito mostrar a correlação das práticas de testes com a geração dos incidentes através do estudo de casos no setor de TI em uma empresa de planos de saúde com atuação em todo Brasil e com sede em Fortaleza-CE.

REFERENCIAL TEÓRICO

Para o referencial teórico deste estudo realizou-se uma pesquisa bibliográfica como embasamento para as respostas que se pretende obter.

Qualidade de software

"Qualidade de software é um processo sistemático que focaliza todas as etapas e artefatos produzidos com o objetivo de garantir a conformidade de processos e produtos, prevenindo e eliminando defeitos" (BARTIÉ, 2002, p. 16).

"A qualidade de software é baseada na tríade processo-tecnologia-pessoas. Esta triangulação envolve tudo aquilo que é relevante na produção de um software. Normas e modelos de qualidade abordam os processos de desenvolvimento, o mercado consumidor de software dita a tecnologia a ser usada, e não há regras ou padrões que tenham uma ênfase específica no ser humano" (SANDHOF, 2004, p. 29).

Segundo Sommerville (2011), a qualidade de um software não depende diretamente da funcionalidade corretamente implementada, mas também dos atributos não funcionais. É necessário estabelecer previamente um conjunto de atributos que estejam relacionados a confiança, usabilidade, eficiência e manutenibilidade. Entretanto, deve-se definir quais desses atributos serão mais importantes para o sistema que será desenvolvido e o processo a ser utilizado para avaliar a qualidade do software, uma forma de verificar a existência dos componentes de qualidade no produto.

De acordo com Bartié (2002), para adquirir qualidade em um software é necessário investir em processos de qualidade durante todos os pontos do desenvolvimento e está ciente de que todas as decisões tomadas durante o curso de elaboração do software serão impactadas na qualidade final do produto.

Metodologias de Desenvolvimento Ágil

Segundo Sommerville (2011), as metodologias ágeis adquiriram espaço dentro das empresas devido à sua maneira de acompanhar as mudanças de acordo com o comportamento do consumidor. Softwares são desenvolvidos rapidamente para que possam suprir as demandas urgentes de negócios, os processos ágeis podem ser caracterizados como uma forma de potencializar as entregas e otimizar os processos.

Os métodos ágeis auxiliam equipes de projetos a lidar com rotinas imprevisíveis e a promover as adaptações necessárias, um projeto pode ser dividido em círculos curtos e entregas constantes, as práticas ágeis promovem a efetividade das entregas de curto prazo e a conquista da qualidade ideal de um produto, alinhando os objetivos de negócio as necessidades dos clientes. A seguir serão descritas de maneira breve as duas práticas ágeis mais utilizadas.

Extreme Programming (XP)

De acordo com Pressman (2011), a metodologia XP baseia-se em um conjunto de valores e princípios que são fundamentais para nortear um projeto de software, são eles: comunicação, simplicidade, feedback, coragem e respeito. O XP proporciona uma flexibilização no desenvolvimento do software que auxilia as equipes de desenvolvimento a terem maior agilidade durante os processos, proporcionando a entrega do produto com a qualidade esperada.

"Os requisitos podem ser expressos como cenários (chamados de histórias do usuário), que são implementados diretamente como uma série de tarefas. Os programadores trabalham em pares e desenvolvem testes para cada tarefa antes de escreverem o código. Quando o novo código é integrado ao sistema, todos os testes devem ser executados com sucesso. Há um curto intervalo entre os releases do sistema" (SOMMERVILLE, 2011, p. 44).

Segundo Pressman (2011) XP adota um conjunto de processos, onde o cliente tem participação colaborativa no seguimento de tarefas realizadas pelos desenvolvedores. O cliente recebe pequenas versões do sistema para validar o que está sendo implementado, permitindo também que sejam detectadas necessidades de alterações dos requisitos. A Figura 1 abaixo apresenta o fluxo desse processo.

Figura 1 - Processo Extreme Programming (XP)



Fonte: Pressman, 2006.

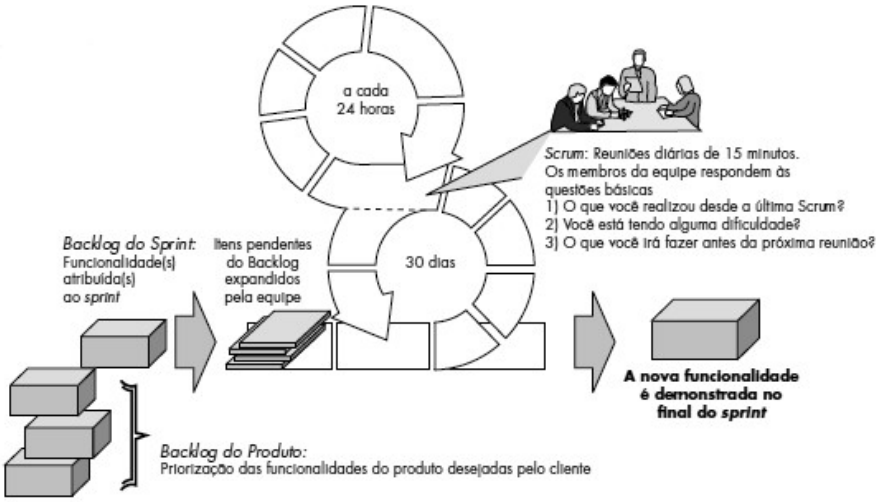
SCRUM

Um ponto importante do Scrum é que o desenvolvimento de software envolve muitas variáveis técnicas e de ambiente, como requisitos, recursos e tecnologia, que podem mudar durante o processo. Isso pode tornar o curso do desenvolvimento imprevisível e complexo, onde requer flexibilidade para acompanhar as mudanças. Espera-se que o resultado do processo seja um software realmente útil para o cliente (MALLMAN, 2011).

"Nas atividades metodológicas, ocorrem tarefas a realizar dentro de um padrão de processo chamado *sprint.* O trabalho realizado dentro de um sprint (o número de sprints necessários para cada atividade metodológica varia dependendo do tamanho e da complexidade do produto) é adaptado ao problema em questão e definido, e muitas vezes modificado em tempo real, pela equipe Scrum" (PRESSMAN, 2011, p. 45).

De acordo com Mallman (2011), as reuniões diárias de acompanhamento destacam-se como uma prática diferencial nesta metodologia, são encontros de curta duração, onde são discutidos pontos daquilo que foi realizado desde a última reunião até a próxima, a proposta é de identificar fatores que cause dificuldades em algum ponto do processo e solucioná-los. A figura 2 abaixo é uma ilustração detalhada de como funciona o fluxo desse processo na prática.

Figura 2 - Fluxo de processo Scrum



Fonte: Pressman, 2011.

Testes de Software

Os testes evidenciam falhas e inconsistências de sistemas, é uma camada importante para os interesses de desenvolvedores e clientes pois mostram a maneira como o software irá se comportar antes ou durante a sua utilização, e se atendem a especificação de requisitos.

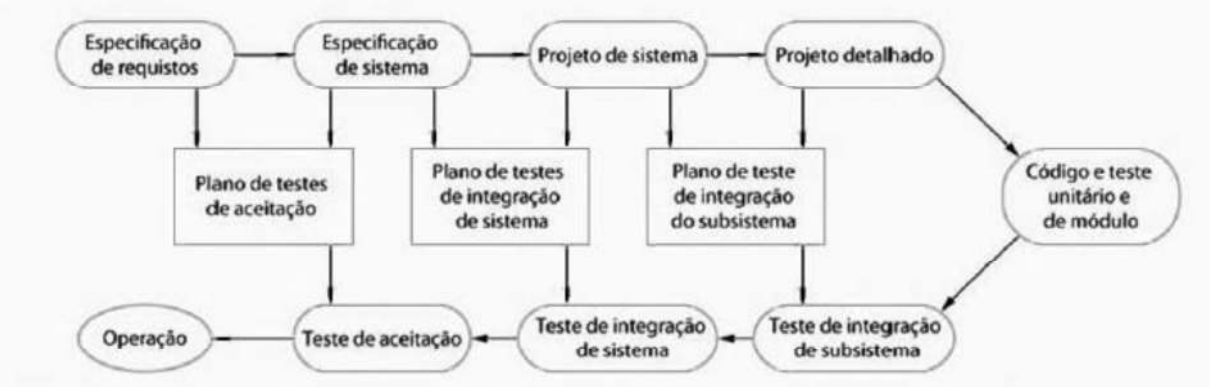
Segundo Bartié (2002), apesar de alguns cronogramas contar com fases específicas de testes, em alguns cenários pode ocorrer a substituição por atividades de correção e manutenção do software. Isso acontece devido à falta de planejamento das atividades de testes, ausência de testes que possam validar funcionalidades antigas ou ausência de um processo de testes automatizados.

De acordo com Pressman (2011), apesar das medidas de esforços adotadas durante a fase de testes, normalmente softwares ainda poderão não funcionar completamente sem ausência de erros, falhas são comuns na programação, e em muitas ocasiões, são resultados de uma complexidade ou negligência por parte dos desenvolvedores durante o processo inicial de elaboração. Por mais eficazes que possam parecer, ainda não há uma alta eficiência em detectar ou remover todos os erros contidos no software, apesar disso parecer desconfortável, são indispensáveis pois aumentam a segurança e a confiança no produto pelo desenvolvedor e pelo cliente.

Adotar uma estratégia de testes deve utilizar testes de baixo nível, necessários para verificar se um trecho de código fonte foi corretamente implementado, assim como os testes de alto nível, que valida as funcionalidades principais de um sistema de acordo com os requisitos do cliente. Essas estratégias devem fornecer orientações ao profissional e definir uma série de metas para o gerente (PRESSMAN, 2011).

Segundo Ammann & Offutt (2008), a documentação é um recurso importante empregado em muitas organizações onde são incluídos planos e relatórios de testes, entretanto focar somente na documentação pode ocasionar em relatórios sem sentidos e sem utilidade. Os planos de testes basicamente contém informações de como os testes foram criados, o porquê de sua criação, como serão executados, informações sobre escopo, recursos e cronograma de atividades. Essa documentação é um requisito essencial implementado em boa parte das empresas. A figura 3 ilustra como um plano de testes pode ser divididos em um cenário de processos de testes.

Figura 3 – Processo de testes de software baseado em planos.



Fonte: SOMMERVILLE, 2011. p. 28.

De acordo com Pressman (2011) optar por um modelo de testes que proporcione acompanhar de perto as rotinas das unidades estruturais de um sistema, contribui para diminuir os riscos de entregar um produto que aparentemente possa vir a apresentar muitas falhas em ambientes de produção. Nesta ocasião, adotar técnicas e planos de testes é importante para rastrear possíveis falhas e antever-se a problemas cada vez maiores que possam ocorrer com o software em produção.

Para que os testes sejam bem-sucedidos, algumas técnicas de testes podem ser empregadas como forma de constatar a presença de erros no software e determinar situações para as quais um software falhe.

Teste funcional (caixa-preta)

Preocupa-se apenas com o comportamento interno do sistema durante a execução dos testes, é indicado para detecção de erros de interface, comportamento e desempenho. Basicamente, ela se importa apenas em checar se as funcionalidades do sistema atendem as expectativas baseando-se nos resultados esperados.

Teste estrutural (caixa-branca)

Permite acesso ao código fonte do programa, conhecendo sua estrutura interna e verificando se os componentes estão acordo com as especificações, sendo assim intitulados como testes de caixa branca. São realizados nas etapas iniciais de desenvolvimento, visando revelar erros lógicos e pressuposições incorretas nos caminhos; trechos executados de forma errônea; e erros aleatórios (FILHO, 2007)

Técnica baseada em defeitos

Escolhe de forma adequada defeitos típicos para criar os casos de teste. A técnica baseada em defeitos usa dados de teste que buscam mostrar a ausência (ou presença) de erros previamente especificados. Por exemplo, para demonstrar que um software trata a divisão por zero corretamente, o dado de teste deve levar à execução dessa divisão por zero. O objetivo é verificar se o programa está livre de defeitos típicos e erros comuns cometidos durante o desenvolvimento.

Testes de unidades

Tem por finalidade testar as unidades internas do sistema de maneira individual e independente com o objetivo de garantir que uma unidade ou classe funcione. Normalmente é de responsabilidade do próprio desenvolvedor da unidade, pois é um resultado de suas próprias experiências.

Testes automatizados

Segundo Sommerville (2011), apesar de muito adotado, os testes automatizados nunca poderão ser completamente automatizados pois verifica-se apenas aquilo que um programa é proposto a fazer, esta técnica pode ser eficaz em alguns casos, mas torna-se limitada quando deseja-se verificar o comportamento de uma interface gráfica de usuário, se a localização dos botões é confortável, checar a facilidade de uso da aplicação, ou testar os efeitos colaterais indesejados de um programa, por exemplo. No entanto, Ammann & Offutt (2008) afirma que os testes de software podem representar um alto custo financeiro e de trabalho quando aplicados a sistemas críticos de segurança, uma alternativa viável a esse tipo de situação é automatizar ao máximo os testes para reduzir custos, minimizar erros humanos e realizar testes de regressão.

Testes de integração

Depois da testagem das unidades, a técnica de integração tem como objetivo verificar e garantir que todos os componentes trabalhem em conjunto de acordo com as especificações descritivas do sistema.

Testes de regressão

Essa técnica é realizada a cada nova versão de um software, onde as funcionalidades são modificadas, ela evita que erros corrigidos antes da nova versão tornem a aparecer quando deseja-se adicionar algo novo.

Testes de usabilidade

Os testes de usabilidade avaliam o grau com o qual os usuários podem interagir efetivamente com o sistema e o grau em que o sistema dirige as ações do usuário, reforçando uma abordagem de interação consistente (PRESSMAN, 2011).

Pressman (2006) afirma que essa técnica permite identificar problemas e observar o comportamento dos usuários durante a utilização do sistema, esses testes são realizados durante a fase de aceitação dos usuários.

Governança de TI

"A governança de TI torna-se cotidianamente mais necessária devido ao processo que motivou seu aparecimento: acelerado desenvolvimento que se caracteriza pela competitividade,

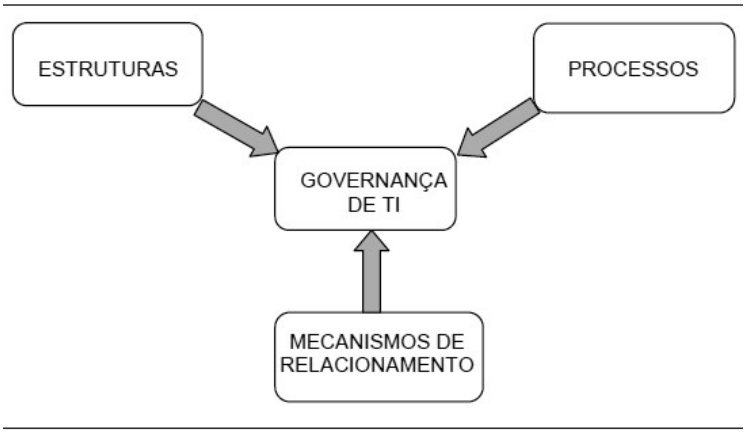
concorrência acirrada, exigência dos consumidores, redução de tempo de vida útil dos produtos e a já citada necessidade de transparência nos processos de negociação." (MORAES, 2010, p. 40).

Segundo Aguiar (2013), a escalabilidade de utilização dos recursos computacionais motivou organizações a adotar medidas de esforço para intervir nas dificuldades enfrentadas pela gestão. A TI deixou de ser algo funcional concentrado apenas em assuntos técnicos e passou a colaborar ativamente nos interesses estratégicos da organização (SCHIAVON; LIMA; PIRES, 2010).

Conceitualmente, a governança de TI pode ser definida como uma forma de garantir com que os papéis e as responsabilidades em uma organização sejam efetivamente desempenhados para assegurar que a gestão e utilização dos recursos de TI contribua com eficiência para o alcance das metas de negócio (CRUZ; ITGI, 2007; SIMONSSON; JOHNSON, 2006, p. 2; WEILL; ROSS, 2006, p. 8).

A governança de TI pode ser dividida em estruturas, processos e relacionamentos. As estruturas são formadas por tomadores de decisão e comitês, responsáveis pelo direcionamento da TI na organização. Os processos refere-se as decisões, as estratégias e acompanhamento dos resultados, o funcionamento da TI em geral. No mecanismo de relacionamento estão incluídos parceiros da TI e demais áreas (PIRES VASCONCELOS; FERNANDES; ABREU, 2012). A figura 4 é uma representação do modelo de Governança de TI.

Figura 4 – Modelo de Governança de TI



Fonte: Moraes, 2010, p. 46

Para Moraes (2010) Aplicar governança de maneira estratégica representa um impacto positivo no desempenho da TI em uma organização; estabelecer políticas, normas e definir processos são determinantes para garantir com que a TI forneça o suporte adequado para ampliar os objetivos e estratégias da organização (SCHWARZ; HIRSCHHEIM, 2003).

CMMI

O CMMI é formado por uma coleção de modelos com as melhores práticas que visam auxiliar organizações com atividades relacionadas a desenvolvimento de produtos e prestação de serviços. O CMMI está em atividade nos mais diversos ramos da indústria, atuando como guia na priorização das ações e melhorias da qualidade dos processos (CARRARETO, 2016).

"O CMMI atua por todo o ciclo de vida do produto ou serviço, partindo da concepção, desenvolvimento e aquisição, até a entrega e manutenção. A Engenharia de Software utiliza este modelo como base de qualidade em desenvolvimento de *software"* (AGUIAR, 2013, p. 48).

De acordo com Bartié (2002) o modelo CMMI é baseado em cinco níveis de maturidade organizacional, onde cada nível pode ser a representação do estágio de maturidade e capacidade alcançado pela organização. Ao atingir um determinado nível, considera-se que a organização atingiu um determinado grau de capacidade.

Os níveis de maturidade do CMMI apontam o desempenho da organização. O nível 1, representa um processo estruturado. Já o nível 2, os projetos são realizados conforme os planos documentados. O nível 3, representa um processo bem caracterizado, entendido e descrito, com melhorias ao longo do tempo. No nível 4, a organização e os projetos adotam informações para gerenciar a qualidade dos projetos. E o nível 5, caracteriza-se por buscar a melhoria contínua do processo visando o alcance dos objetivos de negócio (SILVA, 2011).

ITIL

O ITIL é uma biblioteca de infraestrutura que fornece um conjunto de melhores práticas para gerenciamento de serviços de TI, prover qualidade técnica e eficiência na utilização dos sistemas de informação. Esta biblioteca se firmou como um padrão mundial baseado na experiência de empresas comerciais e governamentais que são cada vez mais dependentes dos recursos de TI (ANDRADE; OGC Aplication Management, 2003).

De acordo com Cestari Filho (2011), a gestão de serviços fornecida pela biblioteca ITIL sugere que a alocação dos recursos de TI disponíveis sejam realizados de forma integrada, de modo que a qualidade seja perceptível aos usuários e clientes, evitando falhas na entrega e operação do serviço. Esse gerenciamento baseia-se por adotar uma conduta proativa para atender as necessidades da organização e contribuir para gerar valor ao negócio.

A aplicação do ITIL proporciona uma maior compreensão dos recursos de TI e uma melhor comunicação dos grupos envolvidos nos processos de TI, direciona todas as partes seguirem uma linha única de entendimento permitindo que um elemento crítico como a comunicação torne-se eficiente (MARTINS, 2006).

Segundo Aguiar (2013), a biblioteca ITIL é formada por cinco publicações onde todas as partes são representadas pelo ciclo de vida do serviço, cada parte exerce uma influência com relação a outra, formando um agrupamento rígido de controle e equilíbrio para que as mudanças sofridas no negócio sejam rapidamente adaptadas aos serviços com eficiência. A figura 5 é uma representação ciclo de vida de um serviço baseado no ITIL V3.

Figura 5 - Ciclo de vida do serviço do ITIL V3.

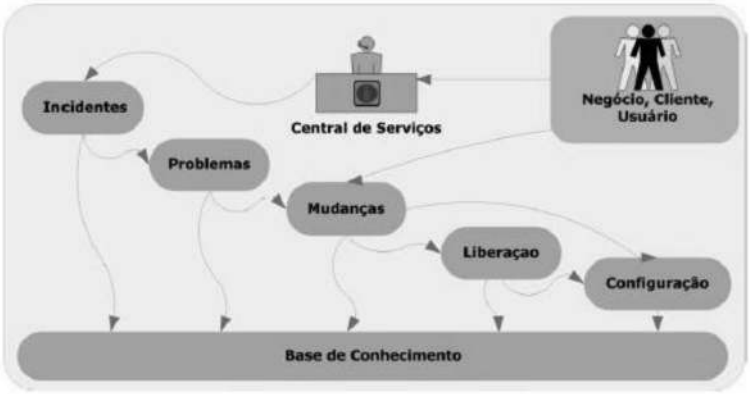


Fonte: Filho, 2012, p. 5.

O ITIL é responsável por melhorar a qualidade dos serviços de TI por meio da otimização dos sistemas e da infraestrutura contando com processos operacionais como forma de gerenciar o ciclo de vida do serviço, assim como compreender as necessidades de clientes e usuários (ANDRADE, 2008).

Martins (2006) afirma que devido ao crescimento das demandas de TI, as áreas responsáveis são obrigadas por fornecer maior qualidade e agilidade no atendimento. As centrais de serviços são responsáveis por atender e resolver chamados, estabelecendo um ponto de comunicação entre os clientes e usuários de TI. A figura 6 ilustra a maneira como os clientes e usuários dos serviços de TI estabelecem contato por meio da Central de Serviços.

Figura 6 - Central de Serviços e as Gerências de Suporte a Serviços de TI



Fonte: MARTINS, 2006, p. 33

Dentre as gerências ilustradas na figura, a partir de agora este estudo focar-se-á na central de Gerenciamento de Incidentes.

Gerenciamento de Incidentes

De acordo com Carrareto (2016) pode-se definir incidente como um evento indesejado ou não esperado que possa impactar a eficiência operacional de uma empresa. Providenciar a eliminação ou obter o controle dos incidentes é uma maneira de antever-se a acidentes, danos aos trabalhadores ou equipamentos, por exemplo (SOBES, 2016).

O gerenciamento de incidentes engloba as demandas de solicitações e eventos que a central de serviços atende, porém não consegue solucionar. Os incidentes são acontecimentos que não pertencem a uma rotina comum de operação dos sistemas e ambientes (MARTINS, 2006).

Segundo Cestari Filho (2011), o processo de gerenciamento de incidentes tem por objetivo buscar a restauração de um serviço o mais rápido possível com o mínimo de interrupções, mantendo os usuários informados da situação; escalonar os incidentes para grupos de atendimentos para que sejam atendidos dentro do prazo e realizar avaliação informando ao processo de gerenciamento de problemas as possíveis causas.

Ao diagnosticar o incidente, há uma sequência de níveis de suporte de serviços a ser recorrido (como representado na figura 6) até que a solução definitiva ou de contorno seja encontrada. Este fluxo é conhecido como escalonamento, e ocorre quando o nível de suporte atual é incapaz de encontrar uma solução, então o incidente é passado para o próximo nível de suporte (CARRARETO; FERREIRA; SILVA, 2008).

"Após o registro do incidente, a Central de Serviços fará sua classificação e tentará resolvê-lo em primeira linha. Havendo a impossibilidade de atendimento, o Gerenciamento de Incidentes irá trabalhar na investigação, análise e recuperação do incidente. O fechamento de um incidente só ocorre mediante a autorização do indivíduo que iniciou o processo" (MARTINS, 2006, p.35). A figura 7 é uma representação do fluxo de incidentes.

Figura 7 – Atividades de gerenciamento de incidentes



Fonte: CESTARI FILHO, 2011, p. 113

METODOLOGIA

Esta pesquisa utilizou uma abordagem de característica qualitativa por ser mais adequada para analisar um determinado setor dos ambientes de TI, para tratar as evidências obtidas através de um questionário estruturado, com perguntas abertas e individuais aos profissionais de TI de uma determinada operadora de saúde suplementar, localizada no município de Fortaleza-CE.

Para este trabalho também foi utilizada pesquisa bibliográfica, possibilitando na construção de um referencial teórico como fonte para direcionamento do trabalho. Desta forma, pretende-se evidenciar as respostas para o problema descrito nesta pesquisa.

A abordagem qualitativa pesquisa detalhadamente os fenômenos do ambiente estudado, o pesquisador vive e conhece a realidade deste grupo ou ambiente. Na pesquisa qualitativa, o pesquisador participa, compreende e interpreta (MICHEL, 2009).

Para delimitar esta pesquisa, foram selecionadas 6 questões relativas aos temas do estudo bibliográfico realizado até aqui. Foi realizado contato telefônico com os profissionais pesquisados, mas devido a pandemia não foi possível o contato pessoal, mas as nuances da pesquisa foi explicada aos profissionais e o roteiro desta pesquisa, conforme o Apêndice D, foi enviado via e-mail para os profissionais responderem ao formulário eletrônico com perguntas abertas no prazo médio de 9,9 dias.

Os profissionais convidados para essa pesquisa foram selecionados pela facilidade de relacionamento do pesquisador com esta empresa que atua no ramo de planos de saúde em todo Brasil com sede em Fortaleza-CE. Essa proximidade facilitou os contatos. A empresa em questão tem uma equipe de TI bem estruturada com mais de 60 profissionais atuando e utilizando as técnicas aqui debatidas.

Cada participante foi convidado formalmente via e-mail, quando foi mencionado que a pesquisa é de caráter científico e acadêmico, registrado por meio de Carta Convite – Apêndice A. Foi comunicado ainda que o nome dos profissionais e o local de trabalho não seriam divulgados, preservando o anonimato dos mesmos, conforme o Termo de Compromisso – Apêndice B.

Foram entrevistados 4 profissionais, 3 analistas de sistemas e 1 gerente de TI, sendo 2 do sexo masculino e 2 do sexo feminino, conforme apresentado no quadro 1 abaixo, que trabalham na região metropolitana de Fortaleza-CE.

Quadro 1 – Tipificação dos profissionais entrevistados

ENTREVISTADO	PROFISSÃO	SEXO	INSTRUÇÃO	TEMPO NA EMPRESA
E1	Analista de sistemas	Masculino	Pós-graduado	6 anos ou mais
E2	Analista de sistemas	Feminino	Pós-graduado	6 anos ou mais
E3	Analista de sistemas	Feminino	Pós-graduado	6 anos ou mais
E4	Gerente de TI	Masculino	Pós-graduado	6 anos ou mais

Fonte: Elaborado pelo autor.

LEVANTAMENTO E ANÁLISE DOS DADOS

Conforme mencionado anteriormente, a entrevista foi realizada através de um formulário online com 6 perguntas abertas com o objetivo de analisar os temas levantados no estudo bibliográfico, cada evidência foi avaliada a partir das respostas dos profissionais, levando em consideração a experiência como fator determinante para enriquecimento das informações coletadas.

Qualidade de Software

Sobre o tema qualidade de software buscou-se identificar a dimensão e os impactos associados ao processo de garantia da qualidade, colocando duas questões em discussão:

Primeira questão: Quais os desafios atuais para garantir a qualidade de um software?

As respostas dos entrevistados apontaram características correlativas a tempo, mudanças no escopo do projeto, comunicação e estrutura organizacional como dificuldades enfrentadas.

> *"Mudança de escopo pode ser considerado um desafio para garantir a qualidade de um produto ou serviço. Uma vez que que este já passou por uma bateria de testes, quando mudanças não previstas ocorrem, acabam por demandar novos testes em um tempo bem menor. Falha na comunicação do que deve ser validado, por exemplo".* ***(E1)***

> *"Um dos grandes desafios é adequar o processo de qualidade/testes de softwares à velocidade que o mercado exige para colocação de softwares na prateleira. Muitos enxergam como mais tempo extra para o projeto, sem considerar os ganhos em diminuição de trabalho de sustentação e aumento na qualidade do software em si".* ***(E2)***

> *"Boa equipe de profissionais, usuários que conheçam bem o negócio e técnicas de gestão de projetos de TI".* ***(E3)***

> *"A definição do processo, o tempo e os custos".* ***(E4)***

Segundo os depoimentos, subtende-se que a indefinição de processos e a falta de planejamentos sejam fatores que possam representar uma tendência nos riscos e problemas citados, impactando assim a qualidade final do produto.

Segunda questão: A qualidade dos testes influencia na satisfação ou insatisfação dos usuários? Por quê?

Com a análise das respostas foi constatado que os testes quando impulsionados com qualidade, são sim motivos de satisfação dos usuários. Deixando claro que quando não realizados da maneira esperada ou simplesmente deixados de lado, podem influenciar a satisfação dos usuários.

> *"Sim. Nenhum usuário gosta de visualizar uma tela de erro, muitas vezes não tratadas. A partir do momento que se entrega um produto ao usuário e ele consegue realizar suas tarefas com eficiência e o sistema consegue responder bem é claro que ficará satisfeito".* ***(E1)***

> *"Um software sem defeitos constantes é um dos motivos de satisfação para os usuários. O usuário precisa da aplicação funcionando e bugs geram um "abandono" do software qualquer que seja a plataforma ou tipo de aplicação".* ***(E2)***

> *"Sim, percebe-se que os usuários notam quando o software foi bem testado ou não. Eles perguntam logo: Como foram os testes de software?".* ***(E3)***

> *"Sim, pois os testes realizados com qualidade garantem uma entrega integra para os clientes, garantindo o nível de satisfação".* ***(E4)***

Cronograma

Sobre o tema cronograma, levantado no questionário, buscou-se trazer evidências que demonstrem a relação do software menos testado e não testado com os prazos de entrega.

Questão: Os prazos de entrega influenciam para que um software seja menos ou mais testado? O que acha disso?

Foi possível perceber que os cronogramas exercem uma influência na testagem do software, o que confirma o questionamento. No entanto o entrevistado 1 relata que isso exercerá uma maior influência quando tratar-se de testes exploratórios que demandam muito tempo e menciona os mecanismos de testes automatizados, onde o prazo não será muito influente.

Já o entrevistado 2 sugere que o tempo para realização de testes deva ser definido no processo de desenvolvimento do projeto. Assim como o entrevistado 3 que sugere algo similar, levantando a hipótese de que as atividades de testes quando descartadas podem impactar em

novos incidentes. O entrevistado 4 relata para a geração de retrabalhos e custos como fatores impactados.

> *"Parcialmente sim, principalmente quando se trata de testes exploratórios, pois demanda um tempo muito maior para dar uma resposta. Por outro lado, há ferramentas que ajuda a automatizar outros tipos de testes e para esses casos, o prazo não irá influenciar".* ***(E1)***

> *"Esse tempo de testes dos softwares já deveria ser computado no processo de desenvolvimento do projeto. Mas a maioria das iniciativas exigem testes rápidos que não são capazes de varrer todo o software, sempre com a desculpa do tempo de entrega".* ***(E2)***

> *"O Cronograma do Projeto deve inserir os testes para que no atropelo o software não vá para produção sem testar, pois algumas vezes esse passo é descartado, mas volta em forma de incidentes".* ***(E3)***

> *"Sim. Muitas vezes para garantir a entrega do software a qualidade e os testes são afetados. e isso pode gerar retrabalhos e custos posteriores".* ***(E4)***

Testes

Sobre os testes, a pesquisa buscou identificar a importância da equipe de testes no software implantado e levantar os critérios utilizados para validar e aceitar os testes.

Primeira questão: Qual é a importância da equipe de testes no software implantado?

De acordo os relatos, é notável a importância da presença de uma equipe de testes no planejamento e desenvolvimento do software, ela contribui por garantir a entrega de um software funcional de acordo com as especificações exigidas e contribuindo para uma menor tendência de defeitos.

> *"A equipe de teste garante que os requisitos funcionais que o usuário solicitou foi atendida".* ***(E1)***

> *"Sem a equipe de testes o volume de defeitos existentes e descobertos pós-implantação é muito maior. Além da equipe contribuir sempre para a boa experiencia do usuário da aplicação".* ***(E2)***

> *"A equipe de Testes acaba dominando também as regras de negócio e como foram implantadas no software, facilitando a relação com os desenvolvedores".* ***(E3)***

> *"De extrema importância".* ***(E4)***

Segunda questão: Como os resultados dos testes são validados? Há critérios para aceitação?

É fácil perceber um padrão para cada tipo de resposta seguindo uma mesma finalidade, é constatado que usuários e clientes são os grandes responsáveis por aprovar os resultados daquilo que foi pré-estabelecido no projeto. Vale ressaltar a opinião do entrevistado 1, em que ele aponta que esses critérios dependem muito do tipo de comportamento de cada empresas e que é necessário saber o que será testado para poder validar.

> *"Depende do comportamento de cada empresa. Existe àquelas que seguem uma metodologia onde para cada software desenvolvido seja baseado no critério de*

> *aceite. No geral, tanto o teste de um produto novo ou correção/melhoria é necessário saber o que deve ser testado".* ***(E1)***
>
> *"Cada demanda, seja de projeto ou sustentação, é testada seguindo os critérios de aceite estabelecidos pelos usuários, seja explicitamente ou pela lógica pré-estabelecida do negócio".* ***(E2)***
>
> *"Os testes são avaliados pela equipe de testes e antes da implantação é feito a homologação com a área usuário".* ***(E3)***
>
> *"Sim. Antes de iniciar as demandas é feito um levantamento desses critérios junto ao cliente para facilitar as validações e o desenvolvimento".* ***(E4)***

Incidentes

Pergunta: Qual a relação da qualidade dos testes com o nível de incidentes após a implantação?

É unanimemente constatado que há sim uma grande correlação dos testes com a quantidade de incidentes gerados pós-implantação do software, deixando claro que quanto maior a abrangência e importância que se dar aos processos de testes antes de ir para a produção, os resultados serão satisfatórios tanto para a empresa quanto para usuários e clientes.

> *"Acredito que quando o teste é feito antes da implantação a incidência de erros e insatisfação do usuário diminui consideravelmente. Isso se os testes realizados consideraram o que o usuário solicitou".* ***(E1)***
>
> *"Quanto maior for a abrangência dos testes menor será o número de incidentes severos, que interrompem a experiência do usuário e diminuem a qualidade perante ele".* ***(E2)***
>
> *"Essas variáveis são altamente correlacionadas, quando os testes são mal feitos ou desprezados, os defeitos voltam em forma de incidentes".* ***(E3)***
>
> *"Creio que a maioria dos incidentes são causados por falhas no levantamento de requisitos e/ou na implementação do código".* ***(E4)***

CONSIDERAÇÕES FINAIS

Este trabalho de pesquisa buscou identificar e compreender a importância da equipe de testes na qualidade do software implantado no setor de TI de uma empresa no ramo de planos de saúde com sede em Fortaleza-CE.

A partir das respostas dos profissionais é possível constatar que em cada tema das perguntas foram apontados fatores relevantes que poderão influenciar na estrutura e no comportamento do software pós implantação.

Como resultado desta análise, este estudo identificou pontos importantes a serem destacados sobre os temas avaliados:

<u>Qualidade</u>

Garantir a qualidade de um software está longe de deixar de ser um desafio, onde a comunicação, tempo, estrutura organizacional e custos são fatores que ainda necessitarão de uma maior atenção para receber melhorias específicas e contínuas ao longo do tempo.

A qualidade na aplicação dos testes também é outro ponto a ser levado em consideração pelas organizações, estabelecer esforços e planejamentos, para que os testes exerçam a influência esperada na satisfação dos usuários com o produto.

Cronograma

Em muitas ocasiões os prazos de entrega são responsáveis por colocar em produção um produto deixando escapar alguma etapa de testes e por optar em adotar testes rápidos sem as condições necessárias para se verificar todo o software. A adesão por ferramentas automatizadas pode ser um bom caminho quando se deseja reduzir custos e minimizar falhas humanas, entretanto essas ferramentas podem se tornar limitadas dependendo do caso.

Testes

Nota-se que as equipes de testes são importantes para assegurar que toda a regra de negócio especificada no projeto seja preservada, estabelecendo uma maior compreensão dos objetivos do software e facilitar a comunicação entre os desenvolvedores, proporcionando uma melhor experiência aos usuários e clientes com o software desenvolvido.

Os critérios de aceite dos testes são fatores que dependerão muito do tipo de metodologia aplicada por cada organização, apesar de não existir um padrão que estabeleça um critério único, na maior parte das ocasiões os ambientes de homologação são responsáveis por delimitar a aceitação de acordo com o que foi definido no escopo do projeto.

Incidentes

De acordo com os relatos, conclui-se que quanto maior a abrangência dos testes realizados antes da implantação menor será a quantidade de incidentes críticos que dificultam a experiência dos usuários, a medida em que muitos erros são fatores correspondentes a documentação e implementação de código.

CONCLUSÕES

Baseado no que foi analisado nesta pesquisa podemos observar que embora as etapas de testes estejam previstas no escopo do projeto, existem ações eventuais e não previstas que podem impactar diretamente o planejamento e a qualidade dos testes e consequentemente levando os sistemas desenvolvidos para os ambientes de produção suscetíveis a receber um maior número de chamados ocasionados por falhas e bugs, elevando os índices de incidentes por cada chamado registrado, ocasionando na insatisfação dos usuários e comprometendo a experiência de usabilidade e a vida útil do software. As conclusões indicam uma forte relação entre o tratamento dado aos testes de software durante as etapas de desenvolvimento com a qualidade final do produto implantado.

TRABALHOS FUTUROS

Este estudo é o início de um esforço de pesquisa qualitativa que visa caracterizar a importância da equipe de testes na contribuição da qualidade do software implantado. Ele

baseou-se apenas em relatos de profissionais experientes que atuam no seguimento e estudos bibliográficos. Porém, ele pode ser estendido a uma pesquisa quantitativa por meio de mineração de repositórios de testes e incidentes. Por exemplo, investigar a correlação dos repositórios de software que armazenam suas respectivas informações para levantar resultados numéricos e representativos que comprove a relação crítica entre essas duas entidades.

REFERÊNCIAS BIBLIOGRÁFICAS

AGUIAR, Wellington Sousa. A governança em tecnologia da informação em empresas de planos de saúde suplementar: estudo de casos no Estado do Ceará. Dissertação (Mestrado em Administração e Desenvolvimento Empresarial) -Universidade Estácio de Sá, 2013.

ANDRADE, Jeime Nunes de. Um processo para implementação das práticas ITIL para gerenciamento de serviços de TI. Dissertação (mestrado) – Universidade Federal de Pernambuco. CIn. Ciências da computação, 2008.

BARTIÉ, Alexandre. Garantia da qualidade de software : adquirindo maturidade organizacional / Alexandre Bartié – Rio de Janeiro : Elsevier, 2002 – 5º Reimpressão.

CARRARETO, Daniel Barbarioli. O processo de gerenciamento de incidentes da ITIL: Estudo de caso de uma empresa do setor de saneamento. Dissertação (mestrado) – Universidade Federal do Espírito Santo, 2016.

CESTARI FILHO, Felício. ITIL v3 Fundamentos/ Felício Cestari Filho. – Rio de Janeiro: RNP/ESR, 2011.

CRUZ, Cláudio Silva da. Governança de TI e conformidade legal no setor público: um quadro referencial normativo para a contratação de serviços de TI. Dissertação (mestrado) – Universidade Católica de Brasília, 2008.

MALLMAN, Paulo Roberto. Um modelo abstrato de gerência de software para metodologias ágeis. Dissertação (mestrado) – Universidade do Vale do Rio dos Sinos, Unidade Acadêmica de Pesquisa e Pós-Graduação, Programa Interdisciplinar de Pós-Graduação em Computação Aplicada, São Leopoldo, RS, 2011.

MARTINS, Márcia Missias Gomes. Gerenciamento de Serviços de TI. Dissertação (mestrado) – Universidade de Brasília, 2006.

MORAES, Gustavo Hermínio Salati Marcondes de Moraes. Indicadores do Uso de TI como suporte à Governança de TI. Dissertação (mestrado) - Escola de Administração de Empresas de São Paulo, 2010.

PAUL AMMANN AND JEFF OFFUTT. Introduction to Software Testing. Cambridge University Press The Edinburgh Building, Cambridge CB2 8RU, UK, 2008.

PIRES VASCONCELLOS, Fabrício. Gestão de Processos de Negócios e Governança de TI: um modelo para avaliação do alinhamento. Dissertação (Mestrado) – Fundação Mineira de Educação e Cultura – FUMEC, 2013.

PRESSMAN, Roger S. Engenharia de software [recurso eletrônico] : uma abordagem profissional / Roger S. Pressman ; tradução Ariovaldo Griesi ; revisão técnica Reginaldo Arakaki, Julio Arakaki, Renato Manzan de Andrade. – 7. ed. – Dados eletrônicos. – Porto Alegre : AMGH, 2011.

SANDHOF, Karen. Fatores Humanos no Processo de Desenvolvimento de Software : Um Estudo Visando Qualidade. Dissertação (mestrado) – Escola Politécnica da Universidade de São Paulo, 2004.

SILVA, Priscila Coelho. Análise de gestão de riscos em projetos de sistemas de informação. Dissertação (mestrado) – Universidade Federal do Rio Grande do Sul, Escola de Administração, Programa de Pós-Graduação em Administração, Porto Alegre, 2011.

SOMMERVILLE, Ian. Engenharia de Software / Ian Sommerville; tradução Ivan Bosnic e Kalinka G. de O. Goncalves; revisão técnica Kechi Hirama. — 9. ed. — São Paulo: Pearson Prentice Hall, 2011.