

Implantação de gerência de configuração em empresa de desenvolvimento de software para gestão pública: um estudo de caso

Configuration management deployment at software development company for public administration: a case study

Aplicación de gestión de la configuración en la compañía de desarrollo de software para la administración pública: un estudio de caso

Jullian Hermann Creutzberg

Mestrando do Curso de Administração pela Universidade do Vale dos Sinos - Minter UNISINOS/UNIDAVI. Especialista em Engenharia de Software pela Universidade do Estado de Santa Catarina - UDESC. Especialista em Gestão de Pessoas pelo Centro Universitário para o Desenvolvimento do Alto Vale do Itajaí - UNIDAVI. Bacharel em Sistemas de Informação pela Universidade para o Desenvolvimento do Alto Vale do Itajaí - UNIDAVI.. Professor do Curso de Sistemas de Informação e Superior de Tecnologia em Design Gráfico na Universidade para o Desenvolvimento do Alto Vale do Itajaí – UNIDAVI, Santa Catarina, Brasil
jullianhc@gmail.com

Nilson Ribeiro Modro

Graduação em Tecnologia Em Processamento de Dados pela Universidade do Estado de Santa Catarina , graduação em Ciência da Computação pela Universidade do Sul de Santa Catarina (mestrado em Engenharia de Produção pela Universidade Federal de Santa Catarina e doutorado em Engenharia de Produção pela Universidade Federal de Santa . Professor associado da Universidade do Estado de Santa Catarina, Santa Catarina, Brasil
nilsonmodro@gmail.com

Luiz Cláudio Dalmolin

graduação em Engenharia Agrônômica - Faculdades Unidas de Bagé mestrado em Ciência da Computação pela Pontifícia Universidade Católica do Rio Grande do Sul e doutorado em Ciência e Tecnologia Ambiental pela Universidade do Vale do Itajaí. Professor efetivo da Universidade do Estado de Santa Catarina, Santa Catarina, Brasil
luiz.dalmolin@udesc.br

Editor Científico: José Edson Lara
Organização Comitê Científico
Double Blind Review pelo SEER/OJS
Recebido em 05.09.2016
Aprovado em 27.10.2016



Este trabalho foi licenciado com uma Licença Creative Commons - Atribuição – Não Comercial 3.0 Brasil

Resumo

Práticas de gerenciamento de configuração aliadas aos processos de gerenciamento de qualidade mostram-se uma solução indispensável no desenvolvimento de software. Este artigo apresenta um estudo de caso em uma empresa que desenvolve softwares para gestão pública e implantou um processo de gerência de configuração associado às mudanças nos processos de gerenciamento de qualidade. Um dos pontos discutidos é em relação aos benefícios e à diminuição nas demandas para correção de problemas no software que, em doze meses, foi de aproximadamente 35%, e o tempo gasto com a correção de problemas, que foi reduzido em 55%. Outro aspecto está associado à inspeção e revisão de software realizadas por meio dos testes de *release*, que propiciam a descoberta de problemas pela própria equipe de testes. Vale destacar que, apesar de realizar algumas propostas visando aperfeiçoar os processos, as mudanças realizadas trouxeram benefícios colaterais à qualidade do software.

Palavras-chave: Gerenciamento de configuração; Qualidade de software; Engenharia de software.

Abstract

Configuration management practices allied to the quality management processes prove to be an indispensable solution in software development. This article presents a case study of a company that develops software for public management and implemented a process of configuration management, associated with changes in process of software quality management. One of the discussed points in this article is about the benefits and the remarkable decrease in the number of requests to fix problems in the software, that in twelve months was approximately 35% and the time spent to correct problems was reduced by 55%. Another aspect is associated with the software review and inspection techniques, conducted by the release tests, that allow problems in the software are found by the test team before getting to the customer. Although some proposals are conducted to further improve processes, the changes have brought many benefits to the quality of software.

Keywords: Configuration management; Software quality; Software engineering.

Resumen

Prácticas de gestión de configuración aliadas a los procesos de gestión de la calidad han demostrado ser una solución indispensable para el desarrollo de software. En este artículo se presenta un estudio de caso de una empresa que desarrolla software para la gestión pública e implementó un proceso de gestión de la configuración asociado con los cambios en los procesos de gestión de calidad. Uno de los puntos de discusión es sobre los beneficios y la reducción de las demandas para solucionar problemas en el software que en doce meses fue de aproximadamente 35%, y el tiempo empleado con la corrección de problemas que se

redujo a 55%. Outro aspecto se relaciona com a revisão do software y la inspección llevada a cabo a través de los ensayos de liberación que permiten el descubrimiento de problemas por el equipo de prueba. Vale la pena señalar que a pesar de realizar algunas propuestas con el objetivo de mejorar los procesos, los cambios realizados han aportado beneficios secundarios a la calidad de software.

Palabras clave: Gestión de la configuración; La calidad del software; Ingeniería de software.

1 Introdução

O processo de desenvolvimento de software está se tornando cada vez mais complexo, seja pelo aumento da complexidade da infraestrutura de tecnologia da informação ou pela dependência das organizações em relação aos serviços dispostos por estas tecnologias (Magalhães & Pinheiro, 2007).

Uma das possíveis abordagens para tentar lidar com a complexidade no desenvolvimento de softwares é a adoção de práticas envolvendo o gerenciamento de configuração, que visa controlar as mudanças, bem como as versões, ou *releases*, entregues aos clientes. Por vezes este processo é considerado parte do gerenciamento de qualidade (Sommerville, 2011).

Pressman (1995, p. 735) afirma que “toda mudança no software tem potencial para introduzir erros ou criar efeitos colaterais que propagam erros”. Portanto é fundamental que o controle de mudanças seja bem definido, aliado ao processo de qualidade de software.

Outro ponto envolvido nesta questão é o gerenciamento de *releases*, que para Sommerville (2011) envolve definir sobre as datas, preparar todas as informações para a distribuição e a documentação de cada versão do sistema. Pode conter ainda uma fase de revisão, inspeção e testes de *release*, visando garantir a qualidade do produto que será entregue aos usuários.

Durante o ciclo de vida de um sistema, inúmeras versões serão desenvolvidas e liberadas aos clientes, sendo os processos de gerenciamento de configuração, aliados ao gerenciamento de qualidade, importantes para a manutenção da vida útil do software. Justifica-se também a adoção destes processos, pois é uma forma de garantir que as mudanças realizadas sejam controladas, previamente analisadas e aprovadas pelos responsáveis, devidamente testadas para que não interfiram negativamente nas funcionalidades já existentes e para que os usuários sejam informados das mudanças ocorridas.

Isto posto, o presente artigo tem como objetivo descrever um estudo de caso realizado em uma empresa de desenvolvimento de software para gestão pública que implantou um novo processo de gerenciamento de configuração, aliado a mudanças nos processos de gerenciamento de qualidade do software.

Para que o objetivo principal possa ser alcançado, este artigo está organizado da seguinte forma: esta introdução, seguida da metodologia da pesquisa. Na revisão teórica serão apresentadas definições sobre gerenciamento de configuração e gerenciamento de qualidade, aliadas aos temas gerenciamento de mudanças, gerenciamento de *releases*, qualidade de software e revisões e inspeções em software.

Em seguida, narra-se como o processo de liberação das versões ocorria na empresa, e quais mudanças foram adotadas, a partir de maio de 2015, no processo de gerência de configuração e qualidade. São ainda apresentados os principais dados do processo, para que seja possível verificar a sua evolução. Os dados apresentados foram cedidos pela empresa objeto do estudo de caso e foram extraídos do sistema de controle utilizado internamente. As informações sobre os processos anteriores e suas mudanças foram levantadas com os gerentes responsáveis pela equipe de desenvolvimento da empresa. Esta coleta de informações ocorreu em março de 2016.

Nas considerações finais do presente artigo apresenta-se uma análise dos benefícios identificados pela empresa conforme os dados apresentados no estudo de caso, bem como se propõem alguns outros procedimentos que podem ser adotados, visando ao aperfeiçoamento do processo. Por fim, são relacionados alguns temas que podem ser explorados a partir da pesquisa realizada, como recomendações para trabalhos futuros.

2 Metodologia

Para a construção deste artigo foi realizada uma revisão de literatura constituída principalmente de publicações em livros, apresentando definições sobre gerenciamento de configuração e gerenciamento de qualidade, bem como sobre os temas conexos: gerenciamento de mudanças, gerenciamento de releases, qualidade de software e revisões e inspeções em software.

Os dados apresentados nesta pesquisa foram cedidos pela empresa, sendo extraídos do sistema de controle utilizado internamente, e as informações sobre os processos anteriores e suas mudanças foram levantadas por meio de entrevistas não estruturadas, com os gerentes responsáveis pela equipe de desenvolvimento da organização. A referida coleta de dados e informações ocorreu no mês de março de 2016.

3 Revisão Teórica

Nesta seção serão apresentadas definições sobre os temas abordados no presente artigo, visando a um melhor entendimento do seu contexto. Primeiramente, serão discutidos os aspectos relacionados ao gerenciamento de configuração e, logo após, os aspectos relacionados ao gerenciamento de qualidade.

3.1 Gerenciamento de configuração

O processo de gerenciamento de configuração define as atividades, mediante aplicação de procedimentos administrativos e técnicos, nas distintas fases do desenvolvimento do software, com o objetivo de identificar e controlar os itens entregáveis de um software e controlar as solicitações e realizações de mudanças, o armazenamento das versões e a distribuição dos itens do software em um *release* (Weber, Rocha, & Nascimento, 2001).

Porém, controlar as mudanças no software e seus efeitos em outros componentes do sistema requer grande esforço. Quanto mais complexa a aplicação, mais componentes estão sujeitos a serem afetados; sendo assim, a gerência de configuração é importante durante o desenvolvimento e é fundamental na fase de manutenção do sistema (Pfleeger, 2004).

Os processos de gerenciamento de configuração são abrangentes e aplicados a todas as atividades de engenharia de software, contudo o gerenciamento de mudanças pode ocorrer a qualquer tempo, conforme afirma Pressman:

É importante fazer uma distinção clara entre manutenção de software e gerenciamento de configuração de software. A manutenção é um conjunto de atividades de engenharia de software que acontece depois que o software é entregue ao cliente e posto em operação. O gerenciamento de configuração de software é um conjunto de atividades de controle e rastreamento que começa quando um projeto de desenvolvimento de software se inicia e

termina somente quando o software é tirado de operação. (Pressman, 1995, p. 917).

Conforme Sommerville (2011) diversas circunstâncias podem requerer mudanças nos sistemas, sejam novas versões de hardware, novas plataformas de sistemas, concorrentes que implementam novas funcionalidades que precisam ser correspondidas, ou ainda novas necessidades dos usuários. Desta forma, “mudanças são feitas para o software e cria-se uma nova versão de um sistema. Portanto, a maioria dos sistemas pode ser pensada como um conjunto de versões, sendo que cada uma delas necessita ser mantida e gerenciada”. (Sommerville, 2011, p. 475).

Já Wazlawick (2013, p. 221) destaca que “uma configuração de software é o estado em que o software se encontra em determinado momento”. Complementa ainda que podem estar relacionados às versões, além dos programas (códigos-fonte), outros documentos eletrônicos que fazem parte do projeto de software. Paula Filho (2003) afirma que projetos e produtos de software sofrem inúmeras mudanças durante o seu ciclo de vida, e que os procedimentos e as técnicas de gestão de configuração contêm artefatos para administrar estas alterações de forma organizada.

Deste modo o gerenciamento de configuração de um produto de sistema de software envolve quatro atividades afins (Sommerville, 2011):

1. Gerenciamento de mudanças: visa manter o acompanhamento das solicitações dos usuários e desenvolvedores por mudanças no sistema, procura definir os custos e o impacto de fazer tais mudanças, bem como decidir se e quando as mudanças devem ser realizadas.
2. Gerenciamento de versões: destina-se a manter o controle das diversas versões do software e assegurar que as mudanças não interfiram umas nas outras.
3. Construção de sistema: processo de desenvolvimento de componentes do sistema, banco de dados e bibliotecas e a compilação e ligação destes elementos para criar um sistema executável.
4. Gerenciamento de *releases*: envolve a preparação de software para a atualização nos clientes e o acompanhamento das versões do sistema que foram liberadas para uso.

Sabendo que as mudanças são inerentes ao processo de desenvolvimento e ciclo de vida de um software, Pressman afirmou: “As mudanças são inevitáveis quando um software de computador é construído” (Pressman, 1995, p. 916). Portanto, é fundamental o controle destas solicitações, bem como das alterações efetivamente realizadas, das versões produzidas e das liberações aos usuários finais do software, sendo ainda muito importante que o usuário seja informado de tais alterações. Pressman mostra ainda que as confusões geradas pelas mudanças entre os envolvidos em um projeto de software ocorrem quando os procedimentos de gerência de configuração não são aplicados, ou seja, as alterações não são analisadas antes de serem feitas, não são registradas antes de serem implementadas, não são relatadas para quem precisa tomar conhecimento ou não são controladas de forma que os erros sejam reduzidos e a qualidade do produto final aumentada (Pressman, 1995).

3.1.1 Gerenciamento de mudanças

“O controle de mudanças é uma atividade procedimental que garante a qualidade e a consistência à medida que são feitas mudanças num objeto de configuração” (Pressman, 1995, p. 937).

Da mesma forma, para Sommerville (2011, p. 478) “o gerenciamento de mudanças destina-se a garantir que a evolução do sistema seja um processo gerenciado e que seja dada prioridade às mudanças mais urgentes e efetivas”.

No controle de mudanças de um sistema de software estarão indicadas as funcionalidades que foram adicionadas, removidas ou modificadas, contendo também os problemas corrigidos e pendências que eventualmente restam para uma versão posterior (Wazlawick, 2013).

As mudanças são inevitáveis no ciclo de vida de um software; possivelmente, não havendo mudanças no sistema, logo ele se torna obsoleto e substituível por qualquer outro mais moderno e que atenda aos requisitos do usuário, conforme afirma Sommerville:

As necessidades e requisitos organizacionais se alteram durante a vida útil de um sistema, *bugs* precisam ser reparados e os sistemas precisam se adaptar às mudanças em seus ambientes. Para garantir que as mudanças sejam aplicadas ao sistema de uma forma controlada, você precisa de um conjunto de processos de gerenciamento de mudanças [...] (Sommerville, 2011, p. 477).

Segundo Pfleeger (2004) são quatro os aspectos principais quando tratamos da evolução dos sistemas: manter o controle sobre as funções do sistema, manter o controle sobre as modificações do sistema, aperfeiçoar as funções aceitáveis já existentes e tomar medidas preventivas para manter o nível de desempenho do sistema. Ou seja, o gerenciamento de mudanças deve se preocupar tanto com o estado atual do sistema, quanto com as mudanças que ocorrem, porém, sem afetar as funcionalidades existentes, mantendo o desempenho em níveis aceitáveis.

O processo de gerenciamento de mudanças “está relacionado com a análise de custos e benefícios das mudanças propostas, a aprovação dessas mudanças que valem a pena e o acompanhamento dos componentes do sistema que foram alterados”. (Sommerville, 2011, p. 478).

Contudo, o gerenciamento de mudança dentro de um contexto de projeto de desenvolvimento de software, pode ainda auxiliar os envolvidos a aceitarem e se adaptarem ao sistema que está sendo modificado sem demasiado estresse. (Dennis & Wixon, 2012).

3.1.2 Gerenciamento de releases

Controlar as entregas realizadas aos clientes, mantendo um agrupamento de informações sobre a versão liberada, faz parte do gerenciamento de *releases*. Para Sommerville (2011, p. 488) “um *release* de sistema é uma versão de um sistema de software distribuída aos clientes”. Da mesma forma, para Wazlawick (2013) um *release*, ou entrega, consiste na distribuição de uma versão do sistema externamente, ou seja, fora do ambiente controlado de desenvolvimento. Complementa ainda que “o controle de versão vai rastrear todos os artefatos do projeto (itens de configuração) e manter o controle sobre o trabalho paralelo de vários desenvolvedores” (Wazlawick, 2013, p. 222).

Verifica-se que em softwares que possuem grande quantidade de usuários, chamados também como softwares de mercado de massa, normalmente é possível verificar dois tipos de *releases*: os *releases* principais, que fornecem novas funcionalidades ao software; e os *releases* menores, que corrigem *bugs* e problemas relatados por clientes (Sommerville, 2011).

Neste sentido, Pressman afirma que “o controle de versão combina procedimentos e ferramentas para gerenciar diferentes versões de objetos de

configuração que são criadas durante o processo de engenharia de software”. (Pressman, 1995, p. 927).

Sommerville (2011) mostra que elaborar e distribuir um *release* é um processo que pode envolver alto custo, pois além dos trabalhos de criação, testes, revisão, documentação, podem estar envolvidos trabalhos de publicidade para convencer os clientes a comprarem a nova versão, ou simplesmente para que eles conheçam as novas funcionalidades atribuídas ao software.

Com a ocorrência de mudanças nos sistemas e com a liberação dos *releases*, *bugs* podem ser gerados e liberados aos usuários. Visando reparar o software, as empresas fornecem correções que geralmente são mantidas em *sites* para que possam ser baixadas pelos clientes. Para Sommerville, o problema relacionado a estas correções que dependem de uma ação do cliente baixar, é que muitas vezes estes desconhecem a necessidade do reparo ou não entendem a necessidade dele. Por este motivo, fornecedores de software de massa geralmente implementam atualizações automáticas em seus sistemas (Sommerville, 2011).

Desta forma, sucintamente, o gerenciamento de *release* consiste no controle das diferentes versões dos componentes do software, bem como no processo de liberação e atualização desta versão no ambiente de produção aos usuários.

3.2 Gerenciamento de qualidade

O gerenciamento de qualidade tem por objetivo verificar se o produto que será entregue ao cliente está consistente com os padrões e com os requisitos, fornecendo uma análise independente do processo de desenvolvimento do sistema. Portanto, a equipe responsável pela garantia de qualidade do sistema “deve ser independente da equipe de desenvolvimento para que eles tenham uma visão objetiva do software. Isso lhes permite reportar sobre a qualidade de software sem sofrer influências de questões de desenvolvimento de software” (Sommerville, 2011, p. 455).

O produto de software que será entregue aos usuários precisa de um controle de qualidade bem definido; neste sentido, Pressman (1995, p. 733) destaca que “a garantia de qualidade é uma atividade fundamental para qualquer negócio que gere produtos que são usados por outros”.

Segundo Weber, Rocha e Nascimento (2001) o processo de garantia da qualidade faz parte dos processos de apoio ao projeto de software, já que auxiliam e

contribuem para o sucesso e qualidade do produto final. Sobre este processo de garantia da qualidade, destacam:

Define as atividades para fornecer a garantia adequada de que os processos e produtos de software, no ciclo de vida do projeto, estejam em conformidade com seus requisitos especificados e sejam aderentes aos planos estabelecidos. [...] inclui questões como garantia da qualidade do produto, do processo e do sistema de qualidade (Weber, Rocha, & Nascimento 2001, p. 5).

Pfleeger afirma que os fabricantes e engenheiros de software buscam maneiras de certificar que o produto que entregam aos clientes está dentro de níveis aceitáveis de qualidade; desta forma devem ser utilizadas estratégias para produção de sistemas com qualidade (Pfleeger, 2004). Neste sentido, a gerência de qualidade procura fornecer estas estratégias, juntamente com artefatos e boas práticas, de modo a assegurar a qualidade dos sistemas entregáveis.

Contudo, os “padrões e processos são importantes, mas os gerentes de qualidade também devem ter como objetivo desenvolver uma ‘cultura de qualidade’ em que todos os responsáveis pelo desenvolvimento de software estejam comprometidos em alcançar um alto nível de qualidade de produto” (Sommerville, 2011, p. 456). Dessa forma, é necessário que as equipes assumam a responsabilidade pelo seu trabalho, independente da fase de desenvolvimento, iniciando com o levantamento de requisitos, passando pela codificação, até os testes e entrega do sistema.

3.2.1 Qualidade de software

A qualidade de software pode ser medida de acordo com a “conformidade a requisitos funcionais e de desempenho explicitamente declarados, a padrões de desenvolvimento claramente documentados e a características implícitas que são esperadas de todo o software profissionalmente desenvolvido” (Pressman, 1995, p. 724).

Afirma Wazlawick (2013) que está incluída na engenharia de software a área de qualidade de software que visa garantir bons produtos, baseados em bons processos, e comenta:

Pode-se falar, então, de dois aspectos da qualidade: a *qualidade do produto* em si e a *qualidade do processo*. Embora não exista uma garantia que um bom processo vá produzir um bom produto, em geral admite-se que a mesma

equipe com um bom processo vá produzir produtos melhores do que se não tivesse processo algum (Wazlawick, 2013, p. 229, grifo do autor).

Os critérios do gerenciamento de qualidade foram convencionados pela indústria manufatureira pretendendo melhorar a qualidade dos seus produtos, de forma que um determinado objeto pudesse ser comparado e avaliado se ele foi produzido conforme as especificações (Sommerville, 2011). Apesar disto, Sommerville afirma que esta abordagem não pode ser completamente adotada para a produção de softwares, pois “pode ser impossível concluir objetivamente se um sistema cumpre ou não suas especificações” (Sommerville, 2011, p. 457).

Neste sentido, algumas razões para esta impossibilidade podem ser consideradas como, por exemplo, a complexidade da engenharia de requisitos, o que leva, muitas vezes, clientes e desenvolvedores a interpretarem os requisitos especificados de formas distintas. Outro ponto são os envolvidos no processo, conhecidos como *stakeholders*, pois, em algumas situações, não é possível atender totalmente as suas solicitações. Desta forma, os *stakeholders* ‘excluídos’ podem enxergar o software com baixa qualidade. Por fim, algumas características podem ser difíceis de medir como, por exemplo, manutenibilidade ou confiabilidade, de forma que seria impossível certificar a qualidade do sistema nestes quesitos (Sommerville, 2011).

3.2.2 Revisões e inspeções

Pressman afirma que “as revisões de software são um ‘filtro’ para o processo de engenharia de software”. Complementa ainda que essas revisões servem para descobrir defeitos durante o desenvolvimento de software para que possam ser eliminados antes da entrega (Pressman, 1995, p. 736).

Segundo Sommerville (2011), no processo de revisão uma equipe revisa o software e a sua documentação em busca de possíveis problemas de não conformidade com os padrões. O objetivo desta atividade é favorecer a qualidade do software e não avaliar o desempenho dos programadores e da equipe de desenvolvimento. Neste sentido, ressalta o autor:

Revisões e inspeções são atividades de controle de qualidade que verificam a qualidade dos entregáveis de projeto. Isso envolve examinar o software, sua documentação e os registros do processo para descobrir erros e omissões e verificar se os padrões de qualidade foram seguidos. [...], revisões e

inspeções são usadas junto com teste de programa, como parte do processo geral de validação e verificação de software (Sommerville, 2011, p. 462).

Pressman (1995, p. 745) destaca: “revise o produto, não o produtor”, com o intuito de reforçar a atividade de revisão como parte do processo de qualidade de software, e não avaliação da produtividade. Para o autor “o benefício óbvio das revisões técnicas formais é a descoberta precoce dos defeitos de software, de forma que cada defeito possa ser corrigido antes do passo seguinte do processo de engenharia de software” (Pressman, 1995, p. 737).

Uma das etapas finais das revisões e inspeções de software, antes da entrega do *release* aos clientes, é o teste de *release*, que consiste em “testar um *release* particular de um sistema que se destina para uso fora da equipe de desenvolvimento. Geralmente, o *release* de sistema é para uso dos clientes e usuários” (Sommerville, 2011, p. 157). Neste sentido, Sommerville enfatiza ainda que uma equipe distinta deve ser encarregada pelo teste de *release*, ou seja, os testes devem ser realizados por pessoal diverso do que participou do processo de desenvolvimento (Sommerville, 2011).

Por fim, Sommerville aponta ainda que “o teste de *release* precisa mostrar que o sistema oferece a funcionalidade, o desempenho e a confiança especificados e que não falhará durante o uso normal”, sendo possível comprometer-se a que o software que será entregue aos clientes cumpra os requisitos mínimos de qualidade aceitáveis (Sommerville, 2011).

4 Descrição do processo de gerência de configuração e qualidade

Considerando as temáticas apresentadas na seção anterior, este estudo de caso relata e demonstra alguns dados no que tange à implantação de gerência de configuração em uma empresa de desenvolvimento de software para gestão pública, que possui sua sede em Santa Catarina. Algumas informações particulares como nome da empresa, clientes específicos e localização física serão omitidas, visando preservar a sua identidade e integridade comercial.

A empresa atua no mercado de desenvolvimento de sistemas exclusivamente para gestão pública há mais de vinte anos, fornecendo softwares para prefeituras, câmaras de vereadores, fundações públicas e autarquias, totalizando atualmente

mais de trezentos clientes em todo território nacional e mais de 150 mil usuários ativos nos seus sistemas.

Seu quadro funcional possui, no ano corrente, mais de trezentos colaboradores diretos, entre equipe de desenvolvimento, suporte ao cliente, implantação e migração, que atuam na criação e manutenção de três produtos de software principais: um exclusivo para a gestão da área da saúde, um segundo para a fiscalização e emissão de notas fiscais eletrônicas de serviço e o terceiro sistema de gestão integrada, que abrange as demais áreas no âmbito da gestão pública. Contudo, apesar de serem equipes de trabalho distintas e serem comercializados separadamente, todos os seus sistemas foram desenvolvidos para trabalharem em conjunto.

O produto de gestão integrada é o principal sistema da empresa, e durante aproximadamente quinze anos foi uma aplicação desenvolvida em linguagem *desktop*, que necessitava de servidores instalados nos clientes para o funcionamento do sistema. Porém, nos últimos cinco anos, buscando a evolução do seu produto, optou-se pelo desenvolvimento do seu software com uma tecnologia voltada para *internet*.

O produto antigo está sendo descontinuado aos poucos, sendo mantido atualmente por uma equipe mínima de aproximadamente seis funcionários, que realizam o suporte ao cliente e a manutenção exclusivamente para correção de problemas e alterações legais, já que há cerca de quarenta clientes que ainda utilizam esta aplicação.

Após um processo intenso de análise, por mais de um ano, optou-se por esta descontinuidade total do sistema antigo e pelo desenvolvimento de um sistema totalmente novo, não sendo reaproveitados os códigos-fonte ou base de dados, somente o conhecimento e as regras de negócio existentes.

Para esta nova aplicação a empresa fornece, como parte do serviço prestado aos órgãos públicos, um servidor dedicado e monitorado para a aplicação e base de dados, sendo responsável também pela guarda redundante das cópias de *backups* e manutenção deste ambiente exclusivo. Desta forma, evita-se a necessidade de o cliente compor uma infraestrutura de tecnologia da informação própria para isto, necessitando apenas de acesso à *internet*.

As informações e os números apresentados a seguir relatam sobre a realidade no processo de desenvolvimento deste novo produto de gestão integrada.

Para dar suporte no processo de desenvolvimento dos seus produtos de software, a empresa dispõe de uma ferramenta própria para controle, que foi produzida internamente. Por meio dela são controladas as versões do código fonte dos programas, os requisitos de software, os modelos de dados, as solicitações de mudança, o suporte ao cliente, a documentação, a publicação dos *releases*, os casos de teste, etc.

No que tange ao controle das versões de código fonte, por exemplo, cada vez que o programador necessita realizar alguma alteração em determinado arquivo ele o requisita ao sistema controlador que realiza automaticamente uma cópia de segurança do mesmo; ao final, no momento em que a alteração é finalizada, uma nova cópia é feita.

O software de gestão integrada da empresa possui sua interface e a base de dados única, em que todos os sistemas trocam informações e realizam transações entre si. Alguns dos sistemas existentes são: tributário, recursos humanos, contabilidade, compras e licitações, sendo cada sistema dividido em diversos subsistemas como, por exemplo, o sistema de recursos humanos é subdividido em folha de pagamento, ponto eletrônico, saúde ocupacional, etc.

As equipes responsáveis por cada um dos sistemas e seus respectivos subsistemas são compostas basicamente por um coordenador, um ou mais analistas de sistemas, alguns programadores, alguns analistas de regras de negócio e alguns analistas de testes, sendo que no geral cada equipe não contém mais do que quinze pessoas.

Pois bem, os processos envolvendo gerência de configuração e gerência de qualidade foram modificados a partir do mês de maio de 2015, visando implantar um maior controle nas mudanças do software, na liberação de *releases*, no processo de garantia de qualidade, nas revisões e inspeções e nos testes de *releases*, para redução dos problemas gerados nos clientes.

Segundo dados da empresa, em março de 2015 se gastavam aproximadamente 66% do tempo produtivo da equipe com correções de problemas no software, inviabilizando qualquer tentativa de novos desenvolvimentos. No final de fevereiro de 2016 este número baixou para aproximadamente 36%. Anteriormente, o processo era realizado da seguinte maneira (conforme Figura 1): com o recebimento da demanda o analista de regra de negócio fazia uma análise da

situação e a repassava ao analista de sistemas para verificar como a mudança seria realizada; este, em seguida, encaminhava ao programador com suas orientações. Após o programador concluir as suas atividades, os arquivos alterados eram repassados para equipe de testes; estes, por sua vez, realizavam os testes com as informações constantes na solicitação de alteração e na documentação gerada pelo programador e analista. Estes testes eram realizados em uma cópia do sistema existente em seu computador totalmente diferente do ambiente do cliente.

Sendo a solicitação atendida e nenhum problema detectado na alteração realizada, os arquivos eram submetidos diretamente para o repositório em produção, e duas vezes ao dia este repositório era replicado para todos os clientes, atualizando automaticamente a versão do sistema com as alterações liberadas.

Este processo na prática era caótico, pois inúmeras vezes o setor responsável pelos testes de qualidade não recebia informações suficientes na documentação gerada pelo desenvolvimento, ou então as alterações não se resumiam simplesmente na solicitação, considerando que alguns programas poderiam estar sendo utilizados em outras rotinas, pois o sistema é totalmente integrado e isto poderia não ter sido testado.

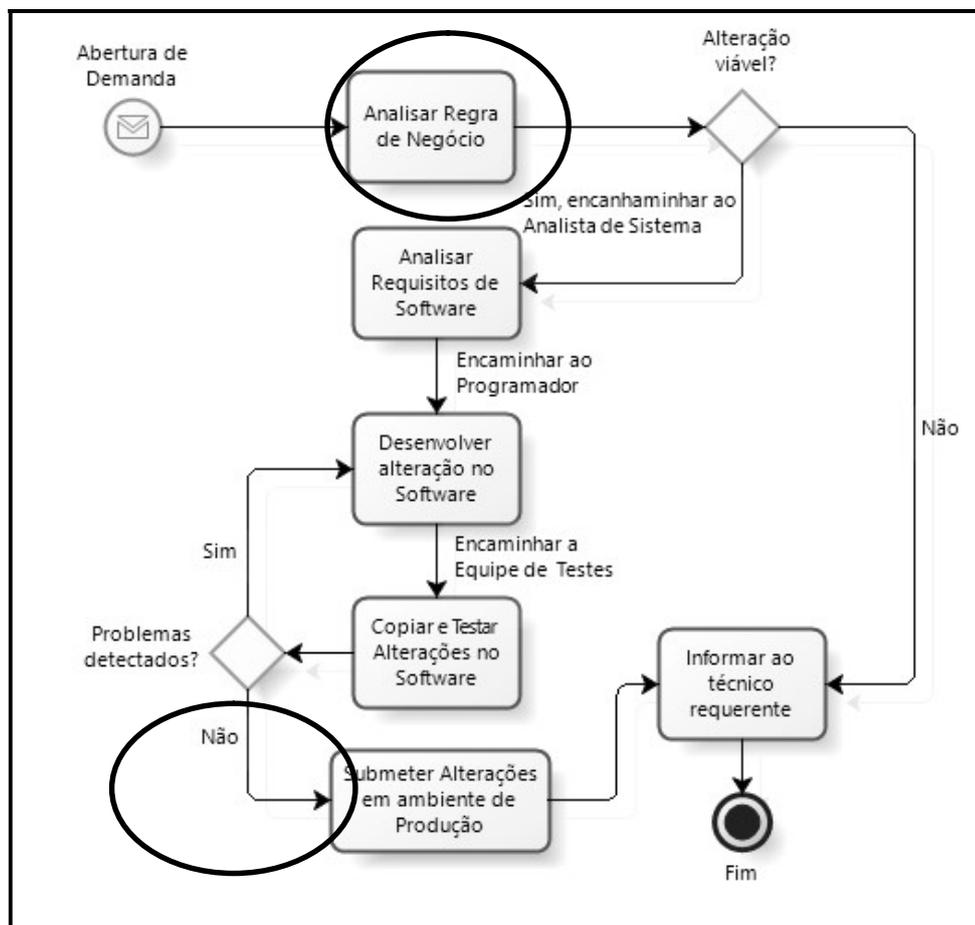


Figura 1

Processo antigo de análise e desenvolvimento das solicitações de mudanças
 Fonte: acervo do autor, 2016.

Visando compreender a evolução do sistema no período de doze meses, a Tabela 1 apresenta algumas informações sobre os sistemas e subsistemas da empresa.

Tabela 1

Informações sobre os sistemas e subsistemas da empresa

Item	Abril/2015	Março/2016	% Variação
Sistemas	06	06	-
Subsistemas	31	48	Aprox. +54%
Requisitos	4.779	7.353	Aprox. +54%
Programas/Arquivos	21.447	26.568	Aprox. +24%
Linhas de código-fonte	1.261.924	1.996.922	Aprox. +58%

Nota: Fonte: acervo do autor, 2016.

Conforme pode ser percebido, em relação à quantidade de linhas de código-fonte houve um aumento de aproximadamente 58%, o que indica também um grande aumento na possibilidade de ocorrência de novos erros.

Após inúmeros estudos, visando à melhoria na qualidade do software da empresa, porém percebendo também a demanda frequente dos clientes pelas melhorias no software, foi iniciado um processo de liberação semanal de *releases*. Dentro do processo de análise e desenvolvimento das solicitações de mudança, o foco principal das melhorias foram nas etapas de análise de viabilidade, priorização das demandas e criação dos casos de teste dos requisitos, conforme os itens circulados na Figura 1.

Neste novo processo (Figura 2), após a primeira avaliação do analista de regra de negócio, um grupo de solicitações é analisado pelo coordenador do sistema, com suporte da gerência e do setor de atendimento ao cliente. Nessa etapa, as demandas são priorizadas e alocadas para desenvolvimento em determinada versão.

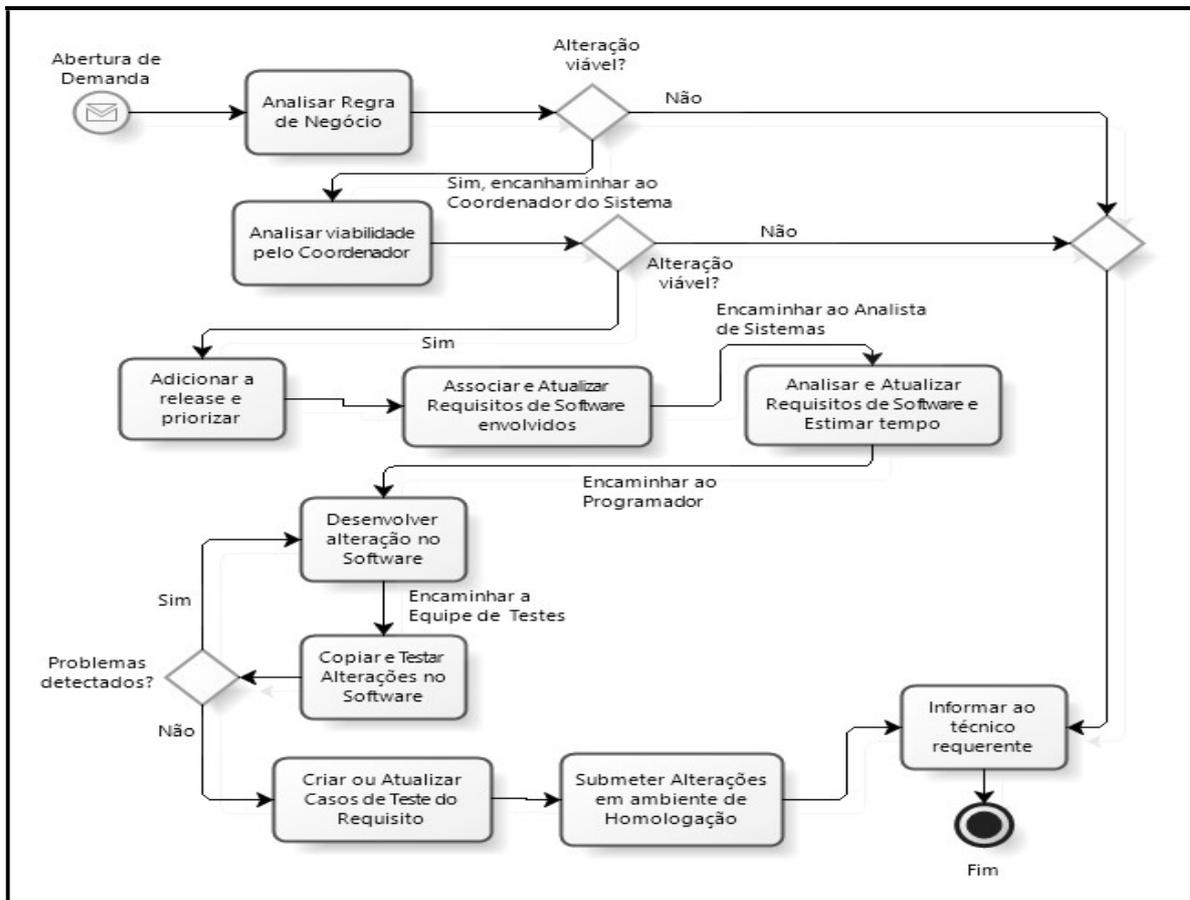


Figura 2

Processo atual de análise e desenvolvimento das solicitações de mudanças

Fonte: acervo do autor, 2016.

Neste momento de análise, a equipe, juntamente com o analista de sistemas, associa cada solicitação de mudança aos requisitos do software que serão alterados ou criados. Cada requisito carrega consigo uma lista de programas que utiliza, ou seja, todos os arquivos que fazem parte daquele requisito devem estar associados a ele. Após definido de que forma será realizada a alteração, os requisitos são atualizados e a demanda é encaminhada ao programador, conforme o tempo definido para atividade e a sua disponibilidade durante a produção do *release*.

Desta forma, o programador somente consegue associar programas para esta solicitação se estiverem relacionados ao requisito. Caso algum arquivo não relacionado ao requisito inicialmente analisado necessite de alteração, o analista de sistemas deve relacionar este novo requisito, garantindo que os requisitos alterados estejam elencados na solicitação de mudança.

As solicitações incluídas na versão depois de desenvolvidas pelo programador são encaminhadas ao setor de qualidade, que por sua vez realiza os testes baseados na documentação gerada pelo programador e pelo analista. O analista de teste tem ainda a responsabilidade de criar um novo caso de teste ou atualizar um existente, que é associado diretamente ao requisito alterado, de como esta alteração no software pode ser testada. Este caso de teste é criado tanto em virtude de uma nova funcionalidade, bem como quando da correção de um problema.

Estes testes, realizados individualmente em cada uma das solicitações, são executados de segunda até quarta-feira a partir de uma cópia dos arquivos em um ambiente especificamente preparado com todas as configurações idênticas ao servidor utilizado pelo cliente. Cada analista de qualidade possui seu próprio ambiente de testes onde, ao realizar a cópia dos programas que estão sendo testados, o sistema de controle faz automaticamente uma atualização de todos os demais programas, conforme a versão mais recente liberada pelas demais equipes.

Após os testes individuais e a criação dos casos de teste, os arquivos são submetidos para um repositório único, onde os técnicos de suporte externo e de atendimento podem realizar novos testes, verificando se a solicitação do cliente foi plenamente atendida ou se ocorrem novos erros após uma determinada correção.

Visando garantir a qualidade da versão a ser entregue aos clientes, foram adotados os procedimentos de testes de *release*, que são realizados durante a

quinta e sexta-feira. Atualmente as equipes de testes de cada sistema possuem de um a quatro integrantes, dependendo da complexidade do sistema relacionado, e estão localizadas na mesma sala onde a equipe de desenvolvimento (analistas e programadores) está trabalhando.

A equipe que realiza os testes de *release* é a mesma que realiza os testes de qualidade no decorrer da semana, porém para estes testes ocorre uma inversão de papéis, de forma que os analistas de testes realizem testes em demandas distintas das que testaram anteriormente.

Para estes testes, primeiramente a equipe de teste realiza uma nova cópia de todos os arquivos a serem liberados na versão semanal para um ambiente de testes único e integrado. Cria-se um plano de testes e todos os casos de testes são executados, de forma que os criados nas outras semanas, quando determinado requisito foi alterado, serão executados antes desta nova liberação com o propósito de que todas as rotinas desenvolvidas anteriormente funcionem plenamente e que nenhum dos erros já corrigidos volte a ocorrer.

Como medida de segurança há um caso de teste mais amplo, onde todas as principais rotinas do sistema são executadas e os principais relatórios são emitidos, visando certificar que as rotinas críticas não sofreram impactos em virtude das alterações no *release* da semana.

Na segunda-feira pela manhã o setor responsável pela atualização realiza a cópia dos arquivos do *release* testados neste ambiente integrado para o ambiente de produção, que em seguida é atualizado em um ou dois clientes, sendo replicado ao final do dia em todos os demais clientes.

O sistema da empresa controla os *releases* liberados, mantendo uma cópia de cada alteração realizada nos programas, obrigatoriamente sendo vinculado a uma solicitação de mudança que visa relatar detalhadamente a alteração. Esta solicitação contém ainda a consideração do técnico que realizou a abertura, do analista de regra de negócio que verificou a demanda, do responsável que autorizou realizar a mudança, do analista de sistema que detalhou ao programador como as mudanças devem ser realizadas e do programador que descreveu o que efetivamente fora realizado, os impactos da alteração no sistema e orientações de testes a serem realizados.

Por fim, o analista de testes também faz suas considerações, com quais testes foram realizados, resumindo o que foi alterado e como pode ser testado, para que

esta informação possa ser repassada a todos os técnicos interessados, servindo de base para documentação do *release* que será enviada ao cliente.

Nesta melhoria no processo, ainda foi inserido o papel do redator técnico, que verifica todas as alterações realizadas que serão liberadas aos clientes no *release*, as documenta e disponibiliza aos usuários de cada sistema.

De forma sucinta, os procedimentos adotados durante os testes dos *releases* e para a atualizações das versões nos clientes são apresentados na Figura 3.

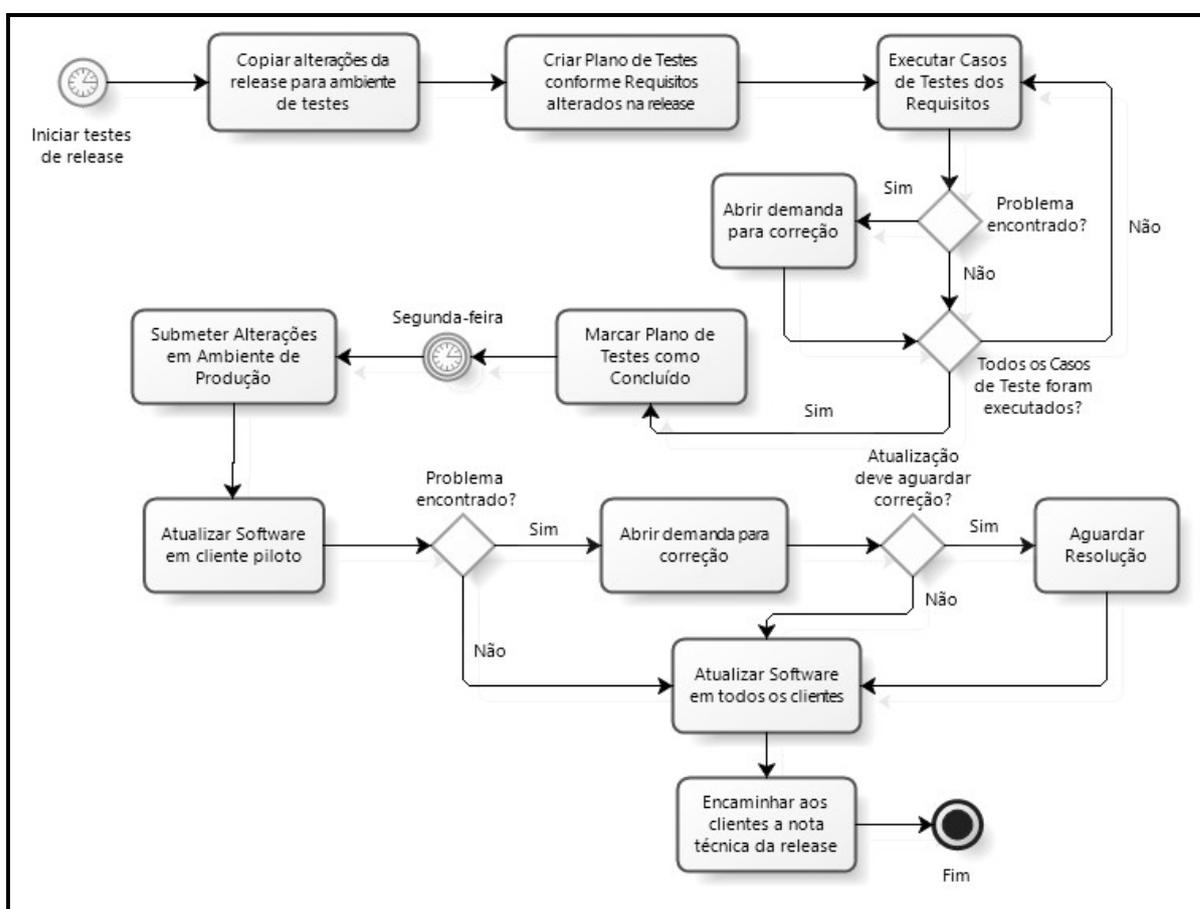


Figura 3
Procedimentos de testes dos *releases* e atualizações das versões
Fonte: acervo do autor, 2016.

Desta forma, com esta alteração no processo de gerenciamento de configuração, dedicando uma atenção especial à qualidade do software, com testes individuais, revisões e inspeções nos programas; testes de *release* integrados em ambientes idênticos aos do cliente; documentação dos casos de teste (para que sejam continuamente executados em cada liberação) e planejamento semanal de

atualizações com o estudo e priorização das solicitações de mudanças e correções, a empresa conseguiu diminuir consideravelmente a demanda pela correção de erros e conseguiu investir mais tempo em melhorias e no desenvolvimento de novas funcionalidades.

A Figura 4 apresenta a quantidade de demandas abertas para correção de problemas em relação ao tempo. Podemos perceber que a quantidade de *bugs* tem uma expressiva variação, considerando o ápice em março de 2015, com 599 demandas abertas. Em fevereiro de 2016 foram abertas 271 demandas.

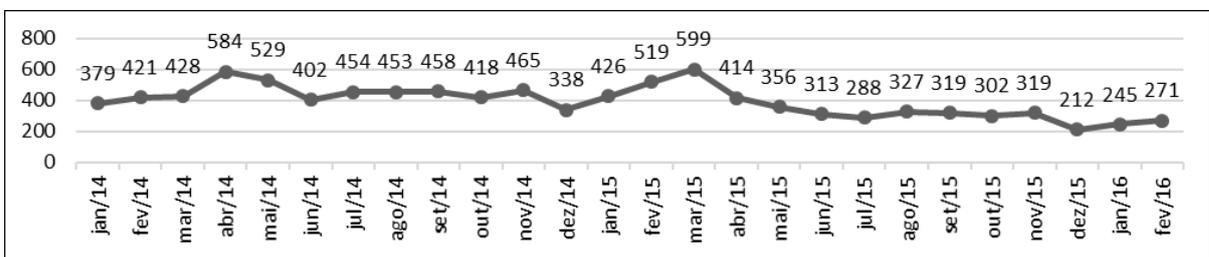


Figura 4

Quantidade de demandas abertas para correção de *bugs*

Fonte: acervo do autor, 2016.

Importante citar que entre abril de 2015 e o início de março de 2016, a empresa teve um crescimento no número de clientes utilizando a aplicação *web* de aproximadamente 63 para 96 clientes.

Foi verificado também que, com os testes de *release*, a própria equipe de testes encontra previamente erros no programa e abre uma solicitação de correção, evitando que o problema seja detectado somente quando o software é utilizado pelo cliente. Desde que este controle foi implementado, em junho de 2015, até o início de março de 2016, aproximadamente 300 novos *bugs* foram encontrados desta forma.

A Figura 5 apresenta a quantidade de demandas liberadas em cada *release*, e a quantidade de demandas liberadas diretamente no ambiente de produção. Estas liberações em produção visam principalmente a correção de *bugs* surgidos após a liberação do *release* ou demandas urgentes.

quantitativos são imediatos. No caso em pauta, a empresa onde se aplicou o estudo de caso obteve em um ano um crescimento de 52% no número de clientes e a melhora de seus processos. Por consequência, cada cliente acaba demandando customizações, visto que, como se trata de sistemas para gestão pública que estão fortemente atrelados à legislação municipal, estadual e federal os mesmos necessitam de alterações conforme a situação de cada cliente.

Gerenciar e manter a qualidade do software é uma atividade árdua que demanda frequente esforço de todos os envolvidos, sobretudo de um sistema utilizado por centenas de clientes. O aumento na quantidade de linhas no código-fonte ou no número de programas é um grande potencial para a inclusão de novos *bugs* nos sistemas. Contudo, mesmo com a expansão dos subsistemas conforme apresentado, verificou-se entre o período de abril de 2015 a fevereiro de 2016 que houve uma redução de aproximadamente 35% no número de demandas abertas para correção de problemas no software.

Decorrente disto houve uma significativa redução no percentual de tempo gasto com a correção de problemas: no período entre março de 2015 e o início de março de 2016, a redução neste tempo foi de aproximadamente 55%.

Desta forma, com a implantação de um novo processo de liberação de *releases*, a adoção do gerenciamento de configuração e o controle de mudanças por um lado a empresa conseguiu diminuir o esforço em corrigir *bugs*, mas principalmente conseguiu aumentar a quantidade de novos desenvolvimentos para seus clientes.

Conforme apresentado, no primeiro *release* semanal a quantidade de correções liberadas era o dobro em relação ao número de novos desenvolvimentos, 80 correções contra 36 melhorias. Por outro lado, no último *release* de fevereiro de 2016, a quantidade de melhorias liberadas foi ligeiramente maior do que a quantidade de correções: 48 correções contra 54 melhorias. Vale destacar ainda que dentro deste processo a definição das prioridades passou a ser mais criteriosa, investindo-se tempo em alterações realmente necessárias e que seriam potencialmente úteis aos usuários.

Com a alteração no processo de gerenciamento de qualidade e testes de software, com a adoção da revisão e inspeção das versões do sistema por meio dos testes de *release*, foi possível detectar problemas antes que fossem liberados aos clientes e identificados pelos mesmos. Porém, enquanto entre junho de 2015 e o

início de março de 2016 foram abertas aproximadamente 2.600 demandas para correções de problemas, somente 11,5% foram abertas pela equipe de qualidade durante estes testes.

Obviamente, apesar da notável diminuição na quantidade de problemas de software, mesmo com o crescimento comercial apontado pela empresa, algumas considerações podem ser feitas visando o aperfeiçoamento do processo de qualidade e gerenciamento de configuração.

Primeiramente, apesar de passar de uma liberação periódica de atualizações para uma liberação semanal de *releases*, é eminente a necessidade de aumentar este período como, por exemplo, versões mensais. Desta forma, pode-se ter mais tempo para o planejamento e realização dos testes, bem como se evita a necessidade de testar semanalmente as mesmas rotinas. Sendo preciso, podem ser planejados *releases* menores, de uma a duas semanas, para liberação de correções de *bugs* ou demandas urgentes.

Em segundo lugar, seria importante que existissem equipes de testes separadas: uma equipe estaria focada nos testes conforme o envio das demandas pelos programadores, e outra estaria voltada somente para os testes do *release*, ou ainda realizando testes independentemente do que está sendo desenvolvido. Além disto, seria importante que o ambiente físico da equipe de testes fosse separado do ambiente de programação, para melhorar a independência das equipes.

Neste quesito, será possível identificar mais problemas gerados pela alteração dos programas da versão antes que sejam liberados aos clientes, ou mesmo apontar *bugs* já existentes que ainda não foram percebidos pelos usuários. Isto visa principalmente diminuir a necessidade de liberação de correções fora dos *releases* programados que, conforme identificado anteriormente, é constante em praticamente todas as semanas.

Outro ponto mencionado em relação aos testes realizados pela equipe para liberação do *release* é a repetição de muitos testes, ou seja, semanalmente as mesmas situações são testadas. Neste ponto poderiam ser criados testes automatizados que agilizariam e facilitariam o trabalho dos responsáveis pela qualidade do software.

Por meio dos processos de gerenciamento de qualidade, com testes automatizados ou realizados manualmente, aliados ao controle e uma análise mais

criteriosa das mudanças no software, podemos diminuir significativamente a quantidade de *bugs* liberados nos *releases* ou encontrados por clientes durante a utilização dos sistemas.

Contudo, podemos concluir que o objetivo deste artigo foi plenamente atingido, mediante a conceituação na revisão teórica sobre os temas correlatos de gerenciamento de configuração e qualidade, bem como com o desenvolvimento da pesquisa e apresentação do estudo de caso. Por meio dos dados apresentados, podemos elencar inúmeros benefícios percebidos pela empresa no que tange ao aumento na qualidade do seu software e no controle das mudanças e versões do mesmo.

Para trabalhos futuros, recomenda-se um acompanhamento contínuo com o intuito de verificar a sua evolução, bem como uma análise dos dados baseada nos conceitos de métricas da engenharia de software. Recomenda-se também a realização de uma pesquisa aprofundada em mais empresas de desenvolvimento de software do estado, visando verificar os processos adotados em gerenciamento de qualidade ou, mais especificamente, na gerência de configuração, na qual seja possível expor um panorama comparativo.

Referências

- Dennis, A., & Wixom, B. H. (2012). *Análise e projeto de sistemas* (2a ed.). Rio de Janeiro: Livros Técnicos e Científicos.
- Magalhães, I. L., & Pinheiro, W. B.(2007). *Gerenciamento de serviços de TI na prática: uma abordagem com base na ITIL, inclui ISO/IEC 20.000 e IT Flex*. São Paulo: Novatec.
- Paula Filho, W. de P. (2003). *Engenharia de software: fundamentos, métodos e padrões* (2a ed.). Rio de Janeiro: LTC.
- Pfleeger, S. L. (2004). *Engenharia de software: teoria e prática* (2a ed.). São Paulo: Makron Books.
- Pressman, R. S. (1995). *Engenharia de software*. São Paulo: Makron Books.
- Sommerville, I. (2011). *Engenharia de software* (9a ed.). São Paulo: Pearson Prentice Hall.
- Wazlawick, R. S. (2013). *Engenharia de software: conceitos e práticas*. Rio de Janeiro: Elsevier.

Weber, K. C., Rocha, A. R. C. da, & Nascimento, C. J. do.(2001). *Qualidade e produtividade em software* (4a ed.). São Paulo: Makron Books.