

Tratamento da Qualidade do Produto e Melhoria Contínua do Processo de Desenvolvimento através de uma Equipe Independente de Gerência de Qualidade

Glaucimar Aguiar, Daniel Paranhos

Objetivos e Justificativa

O projeto apresentado por este artigo consiste na implantação de uma equipe independente da equipe de desenvolvimento, com o objetivo único de primar pela qualidade nos projetos de software através da identificação de problemas no produto, pontos de melhorias nos processos e necessidades de treinamentos nas equipes como meio de prevenir a ocorrência de defeitos.

Esta equipe garante a independência, pois está descomprometida com a solução desenvolvida, mas não com o projeto. Encontrar os problemas é a sua contribuição para o sucesso do projeto. Esta independência é fundamental para o sucesso da atividade.

Para atingir os seus objetivos, esta equipe independente deve monitorar a qualidade dos artefatos gerados durante o processo de desenvolvimento de acordo com técnicas e atributos estabelecidos. Este monitoramento realizado desde os estágios iniciais permite a identificação de questões em aberto ou mal resolvidas no início do projeto, contribuindo para uma maior qualidade e objetividade no atendimento às necessidades do cliente.

Estas atividades trazem efeitos diretos no custo do desenvolvimento e manutenção do software por endereçar aspectos da qualidade de software que são comumente negligenciados e não abordados de forma sistemática. Entre os aspectos tratados de forma sistemática, citam-se: legibilidade de código, facilidade de manutenção, completude e clareza de requisitos, entre outros.

Metodologia de execução

O projeto foi dividido, inicialmente, em três grandes fases: definição, capacitação e execução. A fase de definição é responsável por determinar as atividades das demais fases: em que a equipe precisa ser capacitada e o que fará na execução. O mais importante em todas estas fases é manter o foco no objetivo final: qualidade do produto desenvolvido.

Definição

A fase nomeada Definição consiste em planejar as fases posteriores, pois as decisões tomadas nesta fase determinam as atividades ou especificam detalhes das fases seguintes. Nesta fase, são definidos o processo de

trabalho, as atividades a serem realizadas e o planejamento de início destas atividades. Com base nisso, é possível traçar o perfil da equipe e o que será objeto de estudo para que esta seja capacitada a executar o processo e as atividades planejadas.

A definição do processo de trabalho consiste na primeira atividade, uma vez que as demais dependerão dele. No entanto, o processo não pode ser inteiramente definido em um só passo, o modelo inicial consiste da identificação das várias fases que o compõem e sua descrição. As atividades são definidas logo em seguida.

Para definir o processo de qualidade, é preciso primeiramente estudar o processo de desenvolvimento vigente, uma vez que ambos devem se integrar e comunicar-se através de artefatos. Artefatos gerados no processo de desenvolvimento servem como entrada para o processo da equipe de qualidade. Esta, por sua vez, gera resultados que servem de entrada para o processo de desenvolvimento. Esta interação exige um estudo cuidadoso do processo de desenvolvimento em vigor para que a comunicação entre os dois seja o mais eficiente possível.

Com o macro processo definido, é preciso determinar as atividades a serem realizadas e planejar seu início. Só com esta informação é possível determinar o perfil necessário para a equipe e planejar a sua capacitação.

Neste projeto, o planejamento considerou a ordem de início das atividades como meio de iniciar rapidamente a execução das atividades e melhorar os procedimentos definidos para estas.

As atividades a serem realizadas com o objetivo de melhorar a qualidade dos produtos desenvolvidos foram definidas a partir de pesquisas e estudos em diversas áreas da qualidade e desenvolvimento de software. Estas atividades incluem revisão de requisitos, design, código, técnicas de teste caixa-branca, estudos de desempenho do software e uso de ferramentas de *profiling*.

Capacitação

Com o macro processo definido, as atividades a serem realizadas também definidas e seu início planejado, o perfil da equipe pode ser identificado e o treinamento planejado.

Para o tratamento da qualidade nos níveis abordados neste projeto, os integrantes da equipe de qualidade devem conhecer todas as etapas do processo de desenvolvimento e conseguir realizar as atividades desse processo com excelência. Para isso, é preciso que conheçam os indicadores de qualidade de cada uma das fases e boas práticas de cada uma delas.

De acordo com o planejamento, a primeira atividade a ser realizada seria revisão de código, pois esta seria atividade que apresentaria melhores resultados mais rapidamente. Com isso, a capacitação teve início com boas práticas de programação e destaque especial para as tecnologias utilizadas na instituição. A equipe precisava dominar as tecnologias utilizadas no desenvolvimento para conseguir realizar suas atividades adequadamente e apresentar bons resultados.

De modo geral, os tópicos abordados no treinamento foram: qualidade de software, tecnologias utilizadas na instituição, boas práticas de programação, design orientado a objetos, padrões de projeto, requisitos, qualidade de bons requisitos [4][5][6], testes, técnicas de teste caixa-branca.

Esta fase precisa ser executada periodicamente para reforçar conhecimentos adquiridos e expandir a fronteira do aprendizado. Com isso, o processo de capacitação da equipe torna-se uma tarefa contínua.

Execução

A fase de execução consiste da aplicação dos conhecimentos adquiridos durante a fase de capacitação em atividades que foram definidas na fase de definição. Nesta fase, os guias são construídos ou melhorados e dá-se início ao processo de melhoria contínua do processo e levantamento de necessidades de treinamento dos desenvolvedores.

As primeiras atividades executadas foram as de construção de suítes automáticas de testes utilizando técnicas de teste caixa branca. Em seguida, foram implantadas as análises de *profiling* e análises de desempenho. Estas atividades foram escolhidas por completar as verificações de funcionalidade do produto já realizadas por uma outra equipe: a equipe de testes caixa-preta.

Após o início das atividades de revisão de código, as demais inspeções também tiveram início: revisão de requisitos, design e UI design.

Nesta fase, diversos estudos são realizados para determinar as próximas atividades e contribuir para o crescimento da equipe de desenvolvimento. Vários resultados do projeto são oriundos das análises realizadas nesta fase.

Resultados Relevantes

Ao longo desse projeto, foram desenvolvidas metodologias para auxiliar a execução de algumas atividades da equipe. Essas atividades envolvem a análise da qualidade de alguns documentos como especificação de requisitos, design e design de UI, análise da qualidade do código, além de uma metodologia para a implementação de testes de unidade.

Além desses métodos de análise desenvolvidos, foi necessário também o desenvolvimento de algumas ferramentas, visando facilitar e automatizar o processo de revisão de código que é muito custoso.

Métodos

Para analisar a qualidade dos documentos de requisitos, design e design de UI, foram definidos três guias. Esses guias, juntamente com apresentações, têm o objetivo de capacitar novos membros da equipe a realizarem a atividade e homogeneizar suas análises. Cada documento guia tem uma seção que classifica os possíveis problemas encontrados em quatro categorias: *Defect*, *Minor Defect*, *Nice to Have* e *Open Issue*. Dessa forma, a classificação dos defeitos torna-se bastante consistente, independentemente de quem realizou a análise. Como resultado da revisão, é gerado um relatório (Registro de Revisão) contendo as observações, devidamente classificadas, encontradas durante a revisão.

As revisões geralmente são realizadas por dois ou três membros da equipe, de forma independente, para que diferentes visões sejam utilizadas para capturar a maior quantidade de problemas possível.

Para a realização da atividade de revisão de código, foi definido o Guia de Codificação. Esse guia é composto pelas melhores práticas de programação da linguagem em uso e deve ser obedecido pelos desenvolvedores. O objetivo deste guia é evitar que os defeitos ocorram. Ele possui um conjunto de regras que objetivam prevenir o aparecimento de defeitos.

Durante as revisões de código, percebeu-se a repetição de alguns problemas e que muito tempo estava sendo gasto com o reporte dos mesmos. Assim, surgiu a necessidade da existência de modelos para reportar esses problemas. Esses modelos trazem uma descrição do problema e possíveis soluções. O testador precisa apenas preencher algumas lacunas como a localização do defeito e inserir o texto na ferramenta utilizada para cadastro de problemas.

Na etapa de implementação de testes de unidade também é necessário bastante cuidado. O código desenvolvido deve ser fácil de ser mantido e eficiente na busca por erros. Para alcançar esse objetivo, foi definido o Guia de Codificação de Testes. Esse guia possui uma definição da arquitetura básica a ser seguida por um projeto de testes, como especificar casos de testes, além de uma lista de boas práticas de programação que ajudam na melhoria de sua manutenibilidade.

Esse conjunto de metodologias está trazendo resultados bastante positivos para a equipe de qualidade, em termos de produtividade e qualidade das análises.

Programas de computador

Para reduzir o alto custo de um processo como a revisão de código, foram desenvolvidas duas ferramentas: o *AStyle Visual Studio Add-in* e o *CodeVerifier*.

O *AStyle Visual Studio Add-in* tem o objetivo de formatar o estilo do código fonte de acordo com as regras do Guia de Codificação, funcionando acoplada à ferramenta de desenvolvimento. A outra ferramenta, o *CodeVerifier*, é capaz de detectar automaticamente dois dos problemas mais frequentes encontrados pela equipe de qualidade no código: *Hard-coded Strings*¹ e *Magic Numbers*². O uso dessas ferramentas por parte dos desenvolvedores diminui bastante a quantidade de problemas no código, reduzindo o tempo despendido com a detecção dos problemas, correção e verificação.

Esses produtos estão em vias de serem disponibilizados para o mercado. A intenção é disponibilizá-los de forma gratuita. Dessa forma, pequenas empresas poderão usufruir seus benefícios sem um aumento nos custos, ganhando em competitividade. Ferramentas como o *CodeVerifier*, por exemplo, têm um custo muito alto no mercado, com licenças variando de U\$ 500,00 a U\$ 1000,00.

Aplicabilidade dos resultados e principais impactos na infra-estrutura física da instituição

O resultado da atuação da equipe de qualidade permitirá que decisões sejam tomadas baseadas em critérios objetivos, além de gerar artifícios para ações preventivas e não apenas corretivas. O impacto desse projeto pode ser notado principalmente na qualidade e no custo dos produtos desenvolvidos. A facilidade de manutenção é responsável, também, pela correção e extensão do software de forma mais rápida o que, contribui para atingir os prazos exigidos pelo mercado atualmente competitivo.

Os principais impactos causados pelo trabalho da equipe de qualidade na infra-estrutura física da instituição foram à aquisição de ferramentas para auxiliar na automatização das atividades e contratação de novos integrantes.

Como já mencionado antes, a atividade de revisão de código tem um custo bastante alto e tornou-se necessária a aquisição de uma ferramenta capaz de automatizar a verificação das regras contidas no Guia de Codificação.

¹ Sequências de caracteres encontradas entre aspas no código fonte que não são recursos ou constantes. A presença de *hard-coded strings* dificulta a manutenção do código e não é encorajada como prática de programação.

² Valores numéricos encontrados no código fonte de um software que não são constantes.

Mesmo com toda a preocupação em aumentar a produtividade da equipe, um aumento considerável no volume de trabalho, tornou necessária a contratação de mais três novos membros para a equipe. Esses membros já passaram pelo processo de capacitação e atualmente encontram-se realizando as atividades regulares da equipe.

Características Inovadoras

Este projeto apresenta diversas características inovadoras que se encontram distribuídas em vários aspectos referentes à indústria brasileira, às técnicas de tratamento da qualidade, à organização da equipe e às etapas de envolvimento do time.

Equipes independentes de qualidade na indústria brasileira se popularizaram nos últimos anos, mas ainda não consistem de uma realidade instaurada no país. Quando se investiga mais profundamente estas equipes, descobre-se que o número de equipes que aplicam técnicas de caixa-branca é ainda menor ou inexistente. Mesmo no âmbito mundial, ainda é pequeno o número de empresas que endereçam a qualidade no nível tratado por este projeto. Em virtude desta escassez de exemplos, muita pesquisa e descobrimento foram necessários à execução do projeto para que processos, métodos, técnicas fossem desenvolvidas para atingir os objetivos iniciais. Associado a isso, muito treinamento foi requerido para capacitar os profissionais a atuar nesta área uma vez que mesmo a disciplina de teste ainda é tratada de forma inadequada pelas universidades.

A equipe de qualidade apresentada neste projeto atua desde os estágios iniciais do desenvolvimento do software: os requisitos. *Bell e Thayer* ressaltam que requisitos inadequados, inconsistentes, incompletos ou ambíguos são numerosos e têm impacto crítico na qualidade do software resultante [1]. Estudos confirmam o problema de requisitos em uma escala mais ampla: uma pesquisa feita nos Estados Unidos aponta os requisitos como os grandes responsáveis pelos projetos que não são concluídos ou são concluídos com apenas parte das funcionalidades. Os gerentes apontam as fontes dos problemas: requisitos pobres (13%), requisitos incompletos (12%), alteração nos requisitos (11%), expectativas irreais (6%) e objetivos obscuros (5%) [2]. A atuação nesta etapa do projeto consiste da utilização de inspeções para revisar as especificações geradas como forma de diminuir as inconsistências e ambigüidades com o objetivo de tornar os requisitos mais claros e precisos.

Técnicas de inspeção também são utilizadas nas fases de design e código. A utilização de inspeção nas especificações de requisitos, design e no código é uma prática pouco utilizada na indústria de software nacional especialmente quando realizada por uma equipe independente e treinada

para tal atividade. Revisões de *design* são talvez uma das mais antigas técnicas de depuração, no entanto são também pouco aplicadas [3]. Revisões de design consistem de abordagens que melhoram a “higiene” do produto em desenvolvimento e trazem benefícios significantes a este. Em adição a estes benefícios e às melhorias de legibilidade, estas técnicas sozinhas reduzem o “time-to-market” em 50% [3].

Assim sendo, além da própria existência de uma equipe independente de qualidade, as técnicas e métodos utilizados por tal equipe são inovadoras tanto nacionalmente como mundialmente. Apesar de as técnicas serem aclamadas como detentora de diversos benefícios, a prática destas atividades ainda não é difundida na indústria de software. Adicionalmente, isso contribui para a importância do projeto e o firma como detentor de características inovadoras já que a escassez de estudos e publicações que abordem a prática dos temas aqui tratados exige a busca e, conseqüentemente, a produção de conhecimento e experiência na área.

Conclusão e Perspectivas Futuras

Os resultados deste trabalho demonstraram a importância da equipe independente de qualidade para tratar tópicos que detêm pouca atenção no desenvolvimento de software, mas que são de extrema importância para o atendimento das necessidades dos clientes.

A fase de análise do processo de trabalho definido para a equipe é responsável pelos excelentes resultados alcançados por contribuir na identificação de oportunidades de melhoria da equipe de desenvolvimento promovendo a sua evolução e crescimento. Estas análises são fundamentadas nos dados gerados e nas observações realizadas no dia-a-dia das atividades. É possível comprovar este fato através da observação do número de defeitos encontrados e apontados ao longo da execução das atividades da equipe. As revisões oriundas, por exemplo, de não adequação ao guia de codificação estabelecido decresceram substancialmente desde que as atividades iniciaram, no entanto, a quantidade de código analisado só cresceu.

As revisões de requisitos e design também apresentam melhor qualidade, pois apresentam menos observações e defeitos nas revisões realizadas, reduzindo, assim, o número e tempo de cada ciclo de revisão. As revisões envolvem reuniões, onde os pontos apontados são discutidos e acordados. Estas são as grandes responsáveis pela especialização e crescimento dos autores e, assim, determinantes na melhoria de tais especificações e documentos.

A evolução técnica dos desenvolvedores é uma técnica fundamental no atendimento do objetivo da equipe que consiste em melhorar a qualidade do produto e atender melhor às necessidades do cliente implicando

também na redução dos custos de manutenção e mesmo desenvolvimento do software.

No entanto, a implantação de uma equipe que atua neste nível apresenta várias dificuldades: definição de atividades, processo de trabalho e, principalmente, resistência dos desenvolvedores. Esta última, talvez, a mais difícil de todas. Desenvolvedores, no geral, não estão acostumados com outra pessoa criticando seu código ou a opção adotada para uma solução de forma tão presente e minuciosa. O início das atividades gera muita resistência e deve ser tratada com atenção especial, pois o impacto dessa resistência pode comprometer os resultados alcançados.

A especialização da equipe de qualidade é uma busca contínua, pois, a melhoria do desenvolvimento a força a se especializar cada vez mais e melhores resultados oferecer. Assim, especialização e treinamento da equipe estão longe de ser uma atividade finalizada.

Os resultados alcançados por este projeto são muito valiosos e podem contribuir para outros grupos com atividades similares, sendo assim, a publicação dos resultados aqui apresentados consiste das próximas etapas do projeto como meio de trazer à discussão técnicas e atividades promovendo o engrandecimento do trabalho e contribuindo para a melhoria da própria equipe.

Referências Bibliográficas

- [1] T.E. Bell and T.A. Thayer, "Software Requirements: Are They Really a Problem?", Proc. ICSE-2: 2nd International Conference on Software Engineering, San Francisco, 1976, 61-68.
- [2] The Standish Group, "Software Chaos", http://www.standishgroup.com/sample_research/PDFpages/chaos1994.pdf.
- [3] Bradford L. Goldense, "Implement design reviews the right way". Design News, 1996, página 126. CAHNERS PUBLISHING COMPANY.
- [4] Hooks, Ivy F. "Writing Good Requirements". Proceeding of the Third International Symposium of the INCOSE, Volume 2, 1993. <http://www.complianceautomation.com/papers/writingreqs.htm>
- [5] Kar, Pradip and Bailey, Michelle, "Characteristics of Good Requirements", Proceedings of INCOSE, Volume II, 1996.
- [6] BENDER, R.: Requirements Based Testing. 2004, International Conference on Software Testing Analysis and Review.