

Uma Experiência de Melhoria de Processo utilizando a Análise Causal de Defeitos

Daniela C. C. Peixoto¹, Vitor A. Batista¹, Gustavo M. Rocha¹, Clarindo Isaías P. S. Pádua¹, Rodolfo F. Resende²

¹Synergia – Universidade Federal de Minas Gerais

²Departamento de Ciência da Computação – Universidade Federal de Minas Gerais
Av. Antônio Carlos 6627 - CEP 31270-010 - Belo Horizonte – MG - Brasil

{cascini, vitor, gustavom, clarindo, rodolfo}@dcc.ufmg.br

Abstract. *In this paper we describe a software process improvement experience using the defect casual analysis method. The goal of this approach is to reduce systematic errors during the project development cycle. The identification and analysis of defect causes are used to achieve this goal. In particular, the application of this method has been shown effective for quality improvement and rework reduction.*

Resumo. *Neste artigo relatamos uma experiência com melhoria de processo de software utilizando o método de análise causal de defeitos. Esta técnica tem por objetivo evitar que os defeitos ocorram repetidamente no ciclo de desenvolvimento do projeto a partir da detecção e análise das suas causas. A aplicação deste método mostrou-se efetiva para a melhoria da qualidade e redução do retrabalho no projeto.*

1. Introdução

A crescente demanda por produtos com um alto nível de qualidade força as organizações atuais a adotarem processos que dêem suporte ao desenvolvimento de softwares que atendam os atributos de qualidade requeridos pelos clientes. Uma forma de reduzir o número de defeitos do produto final e, respectivamente, melhorar a sua qualidade, consiste na utilização de processos adequados de verificação e validação incluindo a detecção e prevenção de defeitos [Kan, 2003].

Aliado a esta demanda por qualidade, as organizações de desenvolvimento de software também devem lidar com a crescente competitividade do mercado atual, que exige, cada vez mais, produtos com um maior número de funcionalidades e que sejam entregues em prazos mais curtos. Assim, resta a essas organizações investirem em métodos e técnicas que permitam reduzir o retrabalho e o tempo de desenvolvimento. Neste contexto, a detecção e prevenção efetiva de defeitos têm um papel crucial para alcançar estas metas de prazo, custo e qualidade.

Em particular, técnicas de prevenção e detecção de defeitos como inspeção de software [Gilb & Graham, 1993], medição de defeitos [McGarry *et al.*, 2001] e análise causal de defeitos [Card, 1998] antecipam a detecção dos defeitos e permitem reduzir o custo da sua detecção, já que evitam a propagação deste defeitos para as fases posteriores do desenvolvimento do software, em que o custo de correção é bem mais alto.

No contexto de melhoria de processos, este trabalho caracteriza a utilização do método de análise causal de defeitos. Uma primeira contribuição deste trabalho é uma

discussão da aplicação deste método na Organização dos autores para a prevenção de defeitos durante o ciclo de desenvolvimento de um produto de software. Uma segunda contribuição deste trabalho é a apresentação do impacto na qualidade, nesse escopo, produzido pela implantação das ações resultantes da utilização deste método.

Este trabalho está organizado conforme descrito a seguir. A Seção 2 apresenta de forma sucinta algumas características do processo de desenvolvimento de software utilizado na Organização dos autores. Maiores detalhes do processo podem ser obtidas em trabalhos anteriores, como [Pimentel *et al.*, 2006], [Filho, 2006]. A Seção 3 apresenta um resumo da metodologia de Análise Causal de Defeitos. A Seção 4 apresenta as etapas seguidas para a melhoria do processo. A Seção 5 mostra alguns dos resultados obtidos com a implantação das ações de melhoria de processo e a Seção 6 apresenta as conclusões.

2. O processo de desenvolvimento de software da Organização

A Organização dos autores utiliza uma personalização do processo Praxis [Filho, 2003] de desenvolvimento de software que por sua vez é derivado do Processo Unificado. O processo Praxis está direcionado para o desenvolvimento de aplicativos gráficos interativos, baseados na tecnologia de orientação a objetos.

Na Organização dos autores, as equipes são organizadas por fluxos (ou disciplinas). Os fluxos de Requisitos e Análise foram agrupados em uma única equipe, devido à grande interseção das suas atividades durante os projetos. O mesmo ocorre com Desenho e Implementação. Uma diferença entre o Praxis Padrão e o processo adotado na Organização dos autores consiste na criação da disciplina de Usabilidade.

Uma característica do processo Praxis, mantida na personalização, é a forte orientação a caso de uso, que é dividido em estados, ou marcos de controle durante o projeto. Cada evolução de estado do caso de uso passa por um ou mais procedimentos de controle de qualidade, como inspeções, revisões gerenciais ou testes. Durante a execução desses procedimentos o registro de defeitos injetados no produto é efetuado, para posterior análise. A Tabela 1 lista os estados do caso de uso e os procedimentos de qualidade executados em cada um.

3. Análise Causal de Defeitos (*Defect Causal Analysis - DCA*)

A Análise Causal de defeitos, ou DCA, é uma metodologia de melhoria de processos criada pela IBM na década de 1990. Segundo Card [Card, 1998], o DCA oferece um método simples, de baixo custo para melhoria sistemática da qualidade de software no nível da equipe, projeto ou organização. O principal insumo para o método é o relatório de problemas de software, que é relativamente simples de ser produzido.

O DCA é dirigido por 3 pilares: reduzir defeitos para aumentar a qualidade, usar o conhecimento interno e focar nos erros sistemáticos. Ainda segundo Card, o DCA precisa de poucos pré-requisitos para ser implantado: existência de defeitos, os defeitos precisam estar documentados, a organização deve almejar evitar defeitos em seus projetos e deve documentar e estabelecer o seu processo de desenvolvimento.

A ferramenta recomendada para uso nas reuniões de análise causal é o diagrama de causa e efeito, ou também conhecido como *Ishikawa* ou o diagrama de espinha de

peixe, que procura identificar, de forma estruturada, as causas de cada um dos problemas em análise.

Tabela 1 - Critérios de aprovação dos estados dos casos de uso

Nome	Descrição	Critérios de aprovação ¹
Identificado	Recebeu nome e descrição sucinta.	Inspeção de identificação de requisitos.
Detalhado	Descrição completa dos fluxos e das interfaces de usuário.	Inspeção de detalhamento de requisitos.
Analisado	Os fluxos foram realizados em termos de classes do modelo de análise.	Inspeção de análise.
Desenhado	Os fluxos dos casos de uso de desenho derivados tiveram o desenho externo estático e dinâmico completo.	Inspeção de desenho externo.
Especificado	Os respectivos procedimentos e casos de teste foram completamente especificados.	Inspeção da especificação dos testes.
Realizado	Codificação das classes de fronteira, controle e entidade, além dos testes de unidade	Inspeção de testes de unidade e de código fonte. Execução manual dos testes de implementação pelo implementador (um subconjunto da especificação de testes de integração).
Implementado	A realização dos casos de uso de desenho foi completamente implementada e integrada com o restante do produto.	Inspeção de usabilidade. Re-execução dos testes de implementação pela equipe de testes.
Verificado	Os testes de integração foram executados com sucesso pela equipe de testes e os defeitos encontrados foram todos corrigidos.	Testes de integração manuais, inclusive testes destrutivos.
Validado	Foi aprovado na avaliação dos usuários.	Avaliação pelos usuários.
Completo	As classes que o realizam e possível material correlato foram aprovados em auditoria da qualidade.	Auditoria da qualidade.

Lauesen [Lauesen & Vinter, 2001] descreve um método, bastante similar ao DCA, para a redução do número de defeitos de requisitos detectados nos produtos em desenvolvimento. A principal diferença entre este método e o DCA, é que este método propõe uma avaliação dos defeitos de requisitos por uma equipe que não possua tanto conhecimento da implementação do sistema.

Leszak [Leszak et al., 2000] propõe um método de análise causal da raiz do problema (RCA- *Root Cause Defect Analysis*). Este método é compatível com o DCA. A principal diferença consiste no momento em que ocorre a aplicação do método, ocorrendo apenas após a conclusão do desenvolvimento do sistema.

4. Aplicação do DCA

Os benefícios da utilização do método DCA podem ser observados, principalmente, em termos da melhoria da qualidade e redução do retrabalho. A melhoria da qualidade é obtida ao se eliminar as causas que geram os defeitos. A redução do retrabalho, e, conseqüentemente, a redução dos custos, é obtida diretamente da redução do número de defeitos.

Este trabalho apresenta a aplicação do DCA como um método de melhoria de processos. Este método foi utilizado em um projeto de construção o qual consiste em um sistema WEB que provê informações e serviços para a execução de um processo de

¹ Ver o detalhamento do escopo das inspeções em [Filho, 2003]

compras para um órgão do Governo do Estado. O sistema possui aproximadamente 5.000 Pontos de Função não ajustados e está sendo desenvolvido em plataforma Java por uma equipe de aproximadamente 50 colaboradores.

O DCA foi escolhido por ser um método simples, de fácil implantação e que permite obter melhorias em um curto intervalo de tempo [Card, 2006]. Além disso, foi identificado prontamente o atendimento aos pré-requisitos listados na Seção 3, como descrito abaixo.

- **Processo de desenvolvimento de software definido:** conforme descrito na Seção 2, o processo de desenvolvimento de software da Organização é uma personalização do processo Praxis. Com isto, é possível identificar as atividades nas quais os defeitos foram injetados (local) e o motivo da inserção de tais defeitos (causa).
- **Execução de Inspeções e Testes de software:** devem ser utilizados processos de verificação e validação durante o desenvolvimento do software que permitam a detecção dos defeitos. Neste trabalho, os defeitos foram detectados através de inspeções, que utilizam listas de conferência [Gilb & Graham, 1993], e testes de software. Ambas as atividades são executadas por especialistas, seguindo padrões do IEEE [IEEE, 1997], [IEEE, 1998].
- **Classificação dos defeitos:** todos os defeitos detectados durante o ciclo de desenvolvimento dos projetos foram documentados e classificados de acordo com o padrão do IEEE [IEEE, 1993]. Além disso, os defeitos são registrados indicando o item da lista de conferência associado.
- **Coleta de dados:** após o registro, deve ser realizada a coleta dos dados para a análise e identificação de defeitos recorrentes. Todos os defeitos detectados foram registrados em ferramentas apropriadas.
- **Análise de dados:** a coleta e análise dos defeitos, realizada na organização, seguem a metodologia definida em [McGarry *et al.*, 2001].

Além de atender aos pré-requisitos técnicos, listados acima, um fator crucial para o sucesso da implantação deste processo de melhoria foi o apoio dado pela Diretoria e pelos Gerentes de Projeto. Sem este apoio, não seria possível mobilizar a equipe técnica para participar das reuniões e implantar as ações de melhorias.

As seções a seguir descrevem as atividades executadas no processo de melhoria.

4.1 Identificação de erros sistemáticos

De acordo com Boehm e Basili [Boehm & Basili, 2001], 80% do retrabalho é originário de 20% dos defeitos. Ao se eliminar esses defeitos obtém-se um impacto significativo na melhoria da qualidade e na redução dos custos do projeto [Card, 1998].

A primeira atividade realizada durante o processo de melhoria foi a identificação dos defeitos sistemáticos. Esta atividade foi realizada pelo Grupo de Garantia da Qualidade que utilizou como insumo as informações dos defeitos injetados nos produtos e que foram detectados pelas inspeções, revisões e testes executados durante o ciclo de desenvolvimento do projeto.

Os seguintes questionamentos serviram de base para a seleção dos erros sistemáticos que foram tratados nas reuniões de análise causal. **Tipos de defeitos mais**

recorrentes: este dado foi obtido a partir dos indicadores que apresentavam a distribuição dos tipos de defeitos detectados. **Local dos defeitos:** o local de inserção do defeito e o de sua detecção foram coletados a partir do registro dos defeitos mais recorrentes, permitindo que fosse dado um maior enfoque aos principais aspectos a serem melhorados.

Os procedimentos de qualidade avaliados, e conseqüentemente os defeitos identificados, foram agrupados de acordo com as disciplinas do processo. Utilizando este critério, foi possível distribuir a análise dos problemas entre as diversas equipes, permitindo uma avaliação mais ampla dos fluxos do processo de desenvolvimento utilizado na Organização.

4.2 Seleção dos grupos de análise causal

Os grupos foram divididos de acordo com as disciplinas do processo: Usabilidade, Testes e Desenho & Implementação. Os grupos possuíam 8 pessoas cada, sendo 5 participantes das equipes representativas do fluxo do processo e 3 das demais equipes envolvidas. As pessoas que foram escolhidas possuem conhecimento profundo do processo utilizado na Organização e são especialistas nas suas áreas. Um moderador (representante do fluxo) foi designado para cada grupo.

O motivo deste número reduzido de participantes foi tentar evitar efeitos negativos de reuniões com muitas pessoas, como por exemplo, dispersão e inércia cognitiva [Nunamaker *et al.*, 2003]. A inclusão de pessoas de outras equipes consistiu em uma adaptação do DCA original; o objetivo foi auxiliar na identificação precisa das causas dos problemas e na proposição de soluções que representassem melhoria efetiva do ponto de vista de todos os envolvidos. Com isso, evitou-se que uma melhoria proposta para solucionar problemas específicos de um fluxo (equipe) gerasse impactos negativos nos demais fluxos (equipes). Assim, as análises das causas e soluções seriam avaliadas em todo o contexto da organização.

4.3 Preparação dos participantes

Foram realizadas palestras de apresentação do método de análise causal para toda a Organização e foi elaborado um documento com a descrição detalhada do método, reforçando a motivação para a sua utilização e os benefícios esperados. Esse documento foi distribuído para toda a equipe da Organização. Em seguida, foram realizados treinamentos específicos para os moderadores, que seriam os responsáveis pela condução das reuniões, apresentando em detalhes as etapas a serem seguidas e os problemas já identificados pela Gerência de Qualidade, que deveriam ser tratados durante a reunião. No início de cada reunião, o Moderador respondia a todas as dúvidas que os integrantes ainda possuíam em relação à aplicação do método. A duração das reuniões foi de 2 horas.

4.4 Identificação das causas e propostas de soluções

Cada grupo recebeu um conjunto de 4 problemas para serem discutidos durante as reuniões. A identificação das causas desses problemas foi realizada através da técnica de *brainstorming*. As causas da inserção dos defeitos e os motivos da sua detecção tardia foram estruturados em relação aos fatores de tecnologia, processos, insumos e pessoas. A ferramenta utilizada como apoio para a identificação das causas foi o diagrama de causa e efeito. Cada grupo também identificou um conjunto de ações que

poderiam ser realizadas para evitar a ocorrência dos problemas. Os resultados foram entregues ao Grupo de Ação, que, em seguida, realizaria a priorização das ações e o acompanhamento da sua implantação.

4.5 Priorização e acompanhamento da implantação das ações

O Grupo de Ação é formado por representantes dos grupos de Garantia da Qualidade e de Engenharia de Processos da Organização. As suas responsabilidades consistem em: consolidar as propostas, priorizar as ações, planejar a implementação das ações e alocar recursos, monitorar o progresso da implantação e comunicar os resultados a toda a Organização. As ações foram agrupadas em ações que deveriam ser implantada em um curto prazo (algumas de imediato), médio e longo prazo. Em seguida, foram alocados esforços e designado responsáveis para a execução de todas as ações de curto e algumas ações de médio prazo.

5. Resultados obtidos

Os resultados obtidos com a aplicação do método de análise causal foram medidos após um período de seis meses da sua implantação. De acordo com Card [Card, 1998], embora os benefícios da aplicação do método demorem algum tempo para serem observados, os benefícios da implantação de ações no nível de projeto podem ser obtidas dentro de alguns meses. Foram realizadas 3 sessões de análise da causa dos defeitos. O investimento total para o levantamento e análise dos defeitos foi de 177 PH (pessoas-hora).

As ações de curto prazo (algumas de imediato) foram implantadas, trazendo consigo benefícios. A seguir listaremos alguns dos problemas recorrentes, as causas identificadas pelas equipes técnicas, as soluções propostas e seus benefícios. As Tabelas 2 e 3 exemplificam alguns problemas abordados pelos grupos de Implementação e Desenho respectivamente.

Tabela 2 - Exemplo de problema abordado pelo Grupo de Implementação

Problema	Muitos casos de uso reprovados na re-execução dos Testes de implementação ao passarem pelos procedimentos de qualidade do estado Implementado (ver Tabela na Seção 2).
Causa	Implementadores não executavam corretamente o roteiro de testes antes de entregar o caso de uso para os procedimentos de qualidade e eventualmente faziam pequenos ajustes na execução dos roteiros de testes quando estes possuíam erros ou estavam inválidos ou ambíguos.
Solução	Implantação de uma ferramenta de testes que obriga o Implementador registrar cada um dos passos que executa da especificação de testes. Um <i>log</i> da ferramenta é avaliado pela equipe de testes com o intuito de garantir que o implementador executou esses testes manuais adequadamente, antes de iniciar os testes de integração.
Benefícios	O índice de aprovação nessa inspeção passou de 49% (de um total de 61) para 63% (de um total de 48) após a implantação da ferramenta. Além disso, observou-se uma redução média de 10,15 h/inspeção em retrabalho com correção e verificação de defeitos, totalizando uma economia de 487 h nas 48 inspeções realizadas. O benefício dessa ação correspondeu a 217% do investimento realizado no levantamento e sessões de análise dos defeitos.

Para verificar o efeito das soluções apresentadas, foram medidos alguns indicadores antes e depois da aplicação do DCA. Esses indicadores foram construídos a partir da aplicação da metodologia do *Practical Software Measurement* (PSM) [McGarry *et al.*, 2001]. Os resultados obtidos com a utilização do método DCA podem ser

observados, principalmente, em termos da melhoria da qualidade (eliminação das causas que geram os defeitos) e redução do retrabalho.

Tabela 3 - Exemplo de problema abordado pelo Grupo de Desenho

Problema	Grande quantidade de defeitos de modelagem anotados durante a inspeção de desenho, principalmente com gravidade menor, que poderiam ser evitados.
Causa	Alguns Desenhistas (ou projetistas) não estavam verificando se o caso de uso atendia aos itens mínimos de qualidade exigidos para a liberação do artefato para a revisão. O processo orienta os desenhistas a utilizarem uma lista de conferência preliminar. Entretanto, quando a lista de conferência era utilizada, ocorriam problemas de entendimento (faltavam padrões para alguns comportamentos e descrições), por conta de itens muito genéricos (ou até não aplicáveis).
Solução	Foram atualizadas todas as listas de conferência e materiais de insumo utilizados para a confecção do artefato (desenho do caso de uso).
Benefícios	Após a revisão dos artefatos e listas de conferência e atualização da documentação do processo, observou-se uma redução de 1,45 h/inspeção com retrabalho, o que totalizou uma economia de 50,7 h de retrabalho em 35 inspeções realizadas após as alterações. Outro resultado significativo foi a redução do número médio de defeitos encontrados: <ul style="list-style-type: none"> • <u>Defeitos Críticos</u>: redução média de 0,49 defeitos por inspeção. • <u>Defeitos Maiores</u>: redução média de 3,06 defeitos por inspeção. • <u>Defeitos Menores</u>: redução média de 4,14 defeitos por inspeção.

Um aspecto a ser melhorado na próxima execução do método DCA é a definição dos indicadores a serem monitorados para avaliar os benefícios de cada proposta de ação identificada pelos grupos. Para algumas ações não foi possível identificar os indicadores mais adequados para a avaliação dos resultados, como, por exemplo, para a ação “Promover a conscientização da equipe com o objetivo de utilizar a comunicação informal para divulgação de informações de interesse de todos”.

6. Conclusão

Atualmente, o desenvolvimento de produtos complexos de software é desafiador. O que se observa é que os clientes esperam que os produtos atendam a cada vez mais requisitos de qualidade e, ainda assim, sejam desenvolvidos no mesmo intervalo de tempo ou, até mesmo, em intervalos de tempo mais reduzidos. Assim, desenvolver produtos de software com o nível de qualidade esperado, atendendo aos requisitos de escopo, prazo e custo é uma chave para o sucesso. Uma das formas de atender a essas restrições consiste em utilizar técnicas de prevenção e detecção de defeitos.

Este trabalho relata a experiência de utilização do método de Análise Causal de Defeitos (DCA) com o objetivo de melhorar a qualidade do projeto e reduzir o retrabalho. O DCA mostrou-se uma ferramenta útil para a prevenção de defeitos sistemáticos. Na Organização dos autores este método foi utilizado para evitar que defeitos detectados nos testes e nas inspeções realizadas pelas equipes de Testes, Usabilidade, Desenho e Implementação, fossem novamente inseridos no ciclo de desenvolvimento do produto.

Um fator importante para o sucesso desta iniciativa foi o apoio dado pela Diretoria e pelos Gerentes de Projeto. Outro aspecto, não menos importante, foi envolvimento e o interesse demonstrado pelos participantes dos grupos de análise causal.

Os resultados obtidos com a aplicação do DCA demonstram que após um curto intervalo de tempo já foi possível observar melhorias pontuais na qualidade e na redução do retrabalho.

Com este resultado tão positivo, o DCA foi incorporado ao conjunto de processos da organização e, será aplicado a todos os projetos de desenvolvimento de software de maneira sistemática como parte do processo.

Referências

- [Boehm & Basili, 2001] Boehm, Barry, & Basili, Victor R. 2001. Software Defect Reduction Top 10 List. *IEEE Computer*, vol. 34(Jan), pp.135–137.
- [Card, 1998] Card, David N. 1998. Learning from our mistakes with defect causal analysis. *IEEE Software*, vol.15(no.1), pp.56–63.
- [Card, 2006] Card, David N. 2006 (October). Myths and Strategies of Defect Causal Analysis. In: *Pacific Northwest Software Quality Conference*.
- [Filho, 2003] Filho, Wilson P. Paula. 2003. *Engenharia de Software: Fundamentos, Métodos e Padrões*. LTC.
- [Filho, 2006] Filho, Wilson P. Paula. 2006. Quality gates in use-case driven development. Pages 33–38 of: *WoSQ '06: Proceedings of the 2006 international workshop on Software quality*. New York, NY, USA: ACM Press.
- [Gilb & Graham, 1993] Gilb, Tom, & Graham, D. 1993. *Software Inspection*. Addison-Wesley.
- [IEEE, 1993] IEEE. 1993. *IEEE Standard for Classification for Software Anomalies, Std 1044-1993*.
- [IEEE, 1997] IEEE. 1997. *IEEE Standard for Software Reviews, Std 1028-1997*.
- [IEEE, 1998] IEEE. 1998. *IEEE Standard for Software Test Documentation, Std 829-1998*.
- [Kan, 2003] Kan, Stephen H. 2003. *Metrics and Models in Software Quality Engineering*. second edition. Addison-Wesley.
- [Lauesen & Vinter, 2001] Lauesen, Soren, & Vinter, Otto. 2001. Preventing Requirement Defects: An Experiment in Process Improvement. *Requirements Engineering*, Volume 6 (Number 1).
- [Leszak et al., 2000] Leszak, Marek, Perry, Dewayne E, & Stoll, Dieter. 2000 (June). A case study in root cause defect analysis. In: *Proceedings of the 22nd International Conference on Software Engineering*.
- [McGarry et al., 2001] McGarry, John, Card, David, Jones, Cheryl, Layman, Beth, Clark, Elizabeth, Dean, Joseph, & Hall, Fred. 2001. *Practical Software Measurement: Objective Information for Decision Makers*. Addison-Wesley Professional.
- [Nunamaker et al., 2003] Nunamaker, J. F, Denis, A. R., Valacich, J. S., Vogel, D. R., & George, J.F. 2003. *Group Support Systems Research: Experience from the Lab and Field*. MacMillan Publishing Company.
- [Pimentel et al., 2006] Pimentel, Bruno, Filho, Wilson P. Paula, Pádua, Clarindo, & Machado, Fabiana T. 2006. Synergia: a software engineering laboratory to bridge the gap between university and industry. Pages 21–24 of: *SSEE '06: Proceedings of the 2006 international workshop on Summit on software engineering education*. New York, NY, USA: ACM Press.