

Projeto 6.45

Revisão de Código como Processo Contínuo de Aprendizado e Redução de Defeitos de Software

Alexandre Balestrin Corrêa

Objetivos e Justificativa

Este artigo descreve os resultados obtidos através da implementação de um procedimento de revisão de código em uma equipe de desenvolvimento de software. Também contém as principais dificuldades enfrentadas na implementação de uma política de revisão de código, abrangendo também as vantagens da utilização deste tipo de procedimento em um processo de desenvolvimento de software.

O principal objetivo com a implementação do processo de revisão de código é a redução do número de defeitos. Como objetivos secundários temos a padronização do código, melhoria na clareza do código e melhor facilidade na manutenção.

Metodologia de Execução

Introdução

Nos dias de hoje conhecemos poucas empresas que implementam em seu processo de desenvolvimento a revisão de código (também conhecida como *Peer Review*). Muitos líderes consideram esta metodologia desnecessária, de alto custo e demorada. Este artigo tem por objetivo mostrar que este procedimento contribui e muito para o sucesso dos produtos de software.

Primeiramente gostaríamos de descrever o contexto que o setor de desenvolvimento da Elipse Software estava inserido no final de 2004:

- O sucesso da empresa dependia do software que estava sendo desenvolvido para uso em sistemas críticos de monitoramento de processos industriais;
- Tamanho do projeto: aproximadamente 500.000 linhas em C++;
- A equipe de desenvolvimento era composta por 6 programadores, somente 3 deles estavam desde o início do projeto, que havia sido iniciado em 1999;
- O software já havia sido implantado em algumas dezenas de clientes.

Devido à falta de experiência de parte do time de desenvolvimento, seja nas novas ferramentas ou no novo produto, alguns problemas começaram a surgir:

- Correções em um determinado módulo acabavam gerando um novo problema ou o retorno de um problema já previamente resolvido (regressão);
- O tempo gasto em correções aumentava dia-a-dia;
- Ocorriam falhas devido à má codificação;
- Ocorriam falhas na integração de fontes;
- Havia falta de padronização do código.

Tendo em vista o cenário descrito acima, era necessário fazer uma alteração significativa no processo de desenvolvimento. Foi decidida a implementação de um processo de revisão de código “linha a linha” que teria por objetivo a busca dos defeitos antes que eles fossem para o cliente. Os programadores mais antigos teriam assim a chance de sugerir modificações nas partes de código que tivessem falhas, perdas de performance e má codificação, ou mesmo outras modificações que aumentassem a qualidade do software.

Veremos a seguir como todo esse processo foi evoluindo no decorrer do ano, passando de um processo manual para um bastante automatizado e ágil.

Processo Inicial de Revisão

O Processo Inicial de Revisão tinha os seguintes procedimentos:

- Os fontes da versão atual eram comparados e as diferenças eram montadas em relação aos fontes da versão da semana anterior;
- Uma reunião de revisão de código era realizada com a participação de no mínimo 3 integrantes, com os seguintes papéis:
 - Revisores: todos os programadores presentes na reunião analisavam o código em conjunto, procurando por problemas ou não-conformidades;
 - Anotador: ficava responsável pela anotação das melhorias e problemas encontrados na revisão (chamados de tarefas), assim como por marcar como solucionadas tarefas que tenham sido abertas anteriormente;
 - Operador: ficava responsável pela operação do computador que estava mostrando os arquivos sendo revisados.
- Após a reunião, as tarefas anotadas eram passadas para os demais programadores, para que fossem arrumados os problemas encontrados na revisão. Estas correções seriam revisadas no Processo de Revisão da semana seguinte.

Em conjunto com esses procedimentos, os programadores do Processo de Revisão foram estimulados para que participassem e estivessem motivados para esta mudança que estaria para ocorrer. Foram feitas algumas ações como envolver os programadores na própria definição dos procedimentos a serem adotados, apontando melhorias, ou mesmo distribuição de tirinhas educativas como vemos a seguir.

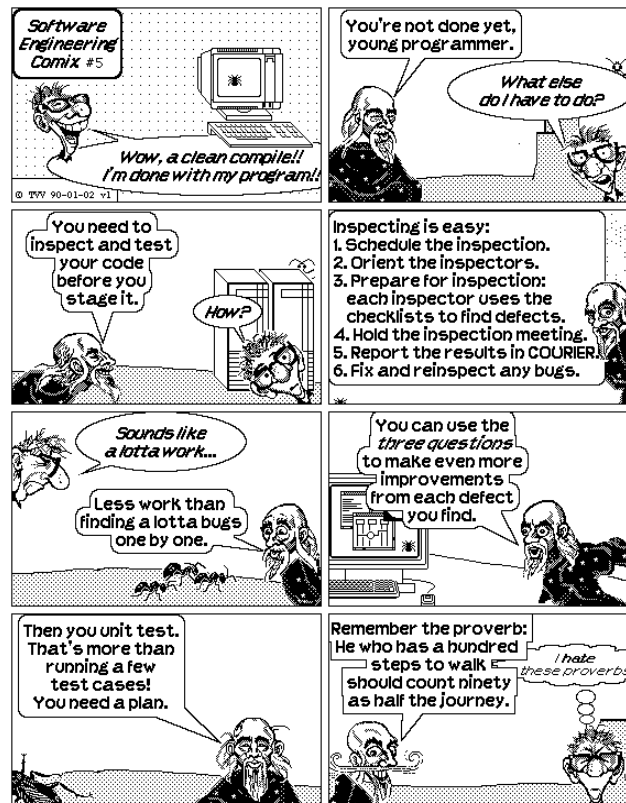


Figura 1. Exemplo de tirinha educativa retirada de <http://www.multicians.org>

Resultados do Processo Inicial de Revisão

Após as primeiras semanas de revisão notou-se uma queda acentuada no número de erros para cada 100 linhas adicionadas, como podemos ver no gráfico a seguir:

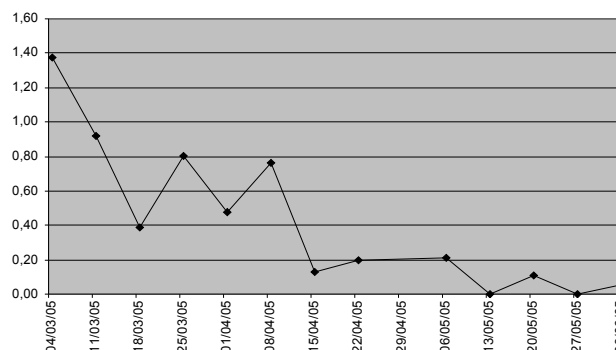


Figura 2. Erros graves encontrados a cada 100 linhas de código adicionadas

Os primeiros 12 meses de revisões tiveram os seguintes resultados adicionais:

Tabela 1. Resultados adicionais do Processo Inicial de Revisão

Número total de reuniões de revisão*	28
Total de horas em revisão	198 h 37 min
Número de erros gravidade ALTA encontrados	41
Número de erros gravidade MÉDIA encontrados	50
Número de erros gravidade BAIXA encontrados	167
Número de outras modificações sugeridas	891

* Cada reunião de revisão era composta de uma ou mais sessões de duração máxima de duas horas, evitando assim o desgaste da equipe.

Note-se que o número de modificações nesse período foi considerável, tanto em relação ao número de erros encontrados quanto ao número de outras modificações sugeridas, como melhorias em performance, padronização de código, estilo, embora o foco dado à revisão de código seja sempre um: procurar por erros.

Problemas do Processo Inicial de Revisão

Apesar dos resultados positivos terem superado as expectativas iniciais, alguns problemas relacionados ao processo começaram a aparecer:

- Logística: alguns recursos eventualmente não estavam disponíveis, como a sala de reuniões e retroprojektor. Em torno de uma hora era gasta preparando o computador do Operador (que tinha que conter as diferenças dos arquivos fontes);
- Atrasos nos prazos, pois as estimativas de término das tarefas ainda não previam o tempo necessário para as revisões;
- Muitos programadores entraram em conflito por acreditarem que

uma crítica ao seu código era pessoal. Os gerentes do processo devem deixar claro que o importante é a melhoria do código e a procura por erros, adotando práticas que de código “defensivo”; portanto, preferências pessoais tem que necessariamente serem deixadas de lado;

- Durante o processo teve que ser criado um documento base que definisse quais seriam os padrões de codificação, evitando assim discussões e defesas de estilos próprios, e garantindo que mesmo com equipes de revisão diferentes, o resultado da revisão fosse padronizado;
- A revisão se mostrou muito mais complicada em códigos-fonte que nunca haviam sido revisados, visto que se encontravam com práticas diferentes das adotadas nas revisões. Estes arquivos foram marcados e seus códigos padronizados.

Notamos que para a continuarmos com a revisão de código precisaríamos de ferramentas de apoio que pudessem proporcionar maior rapidez no processo, assim como uma rastreabilidade superior. Na época, nenhuma ferramenta se mostrou satisfatória. Assim, foi desenvolvido internamente um sistema de revisão automático que detalharemos a seguir.

Processo Automatizado de Revisão (MelloReview)

Considerando os motivos anteriormente descritos no Processo Inicial de Revisão, criamos então a ferramenta MelloReview, com as seguintes características:

- O sistema monta as diferenças do repositório de fontes automaticamente, cadastrando cada modificação na lista de revisões pendentes para cada revisor;
- O revisor utiliza seu próprio computador, para revisar o código, sem a necessidade de haver uma reunião formal, fazendo-a no momento que achar mais conveniente;
- As tarefas abertas entram na lista de tarefas pendentes para o programador que incluiu a modificação;
- Quando o programador resolve uma tarefa aberta, esta modificação entra automaticamente na lista de revisões de tarefas resolvidas para os demais revisores;
- Emite um relatório gerencial contendo as revisões feitas e pendentes, assim como as tarefas a serem resolvidas para cada programador.

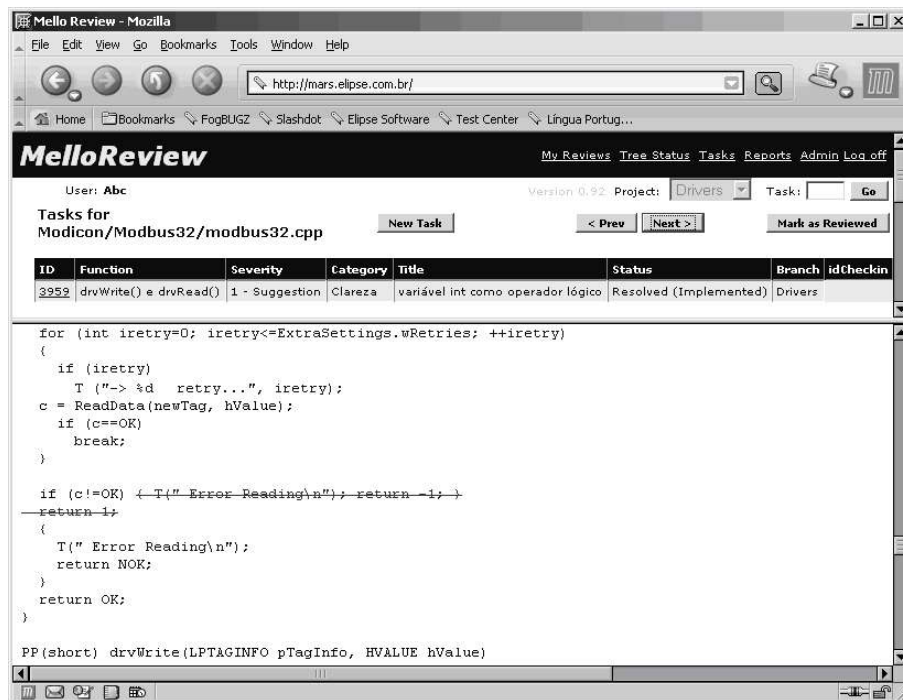


Figura 3. Interface do MelloReview, ferramenta utilizada para a revisão

Resultados do Processo Automatizado de Revisão

Depois de implementada a ferramenta MelloReview, o processo de revisão ficou muito mais dinâmico e confiável:

- Maior agilidade na revisão pois com o uso da ferramenta ela pode ser feita a qualquer momento e independentemente dos demais revisores, que podem estar ocupados com outras tarefas ou mesmo ausentes;
- Melhor rastreamento das modificações sugeridas durante a revisão, assim como a verificação de quais mudanças foram ou não efetuadas;
- Melhor previsibilização dos prazos de lançamento de uma nova versão.

Tabela 2. Resultados do Processo Automatizado de Revisão em 21 meses

Número total de revisões em fontes	59.743
Número de erros gravidade ALTA encontrados	123
Número de erros gravidade MÉDIA encontrados	447
Número de erros gravidade BAIXA encontrados	167
Número de outras modificações sugeridas	4.311

Problemas do Processo Automatizado de Revisão

Acreditamos que os problemas mais críticos do processo de revisão foram solucionados com a implementação da ferramenta MelloReview, porém alguns detalhes ainda tem que continuar na mente dos revisores e líderes:

- Por ser descentralizado e não existir mais o conceito de “reunião para a revisão”, existe uma tendência natural ao processo de revisão ser deixado em segundo plano. Cabe ao responsável pela revisão monitorar e agir para que os revisores não abandonem a prática da revisão em momentos de crise. Somente lançar uma nova versão do produto quando toda a revisão tiver sido efetuada.
- Necessitam ser respeitadas as diferenças de cada revisor. Cada um tem sua velocidade e capacidade de achar problemas, levando tempos diferentes para efetuar suas revisões. Revisores com pouca ou nenhuma experiência tendem a precisar de tempo para começarem efetivamente a encontrar falhas ou mesmo escrever código mais seguro.

Resultados Relevantes

Além dos resultados já apresentados no item anterior, podemos citar também:

Métodos e/ou algoritmos desenvolvidos: como descrito no item anterior, foi desenvolvida uma metodologia para a implementação de um processo de revisão de código, que é composta do processo e do software “MelloReview” (desenvolvido internamente) como ferramenta de apoio.

Eventuais parcerias ou programas de transferência de tecnologia efetuados: Os procedimentos e o software foram apresentados a cinco outras empresas, que demonstraram grande interesse em implantar sistemas de revisão baseadas nessa abordagem.

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

Os impactos na infra-estrutura da instituição foram nulos, visto que todos os recursos já existiam e foram utilizados da mesma forma que antes. Foi apenas necessário configurar um servidor já existente e fazê-lo executar o

módulo servidor do software MelloReview.

Características Inovadoras

Na época que o software foi desenvolvido, não era possível encontrar algum software de mercado que fizesse o processo completo de revisão. Acreditamos que esse software traz uma grande inovação, tanto pelos resultados que gera quanto à sua simplicidade de operação.

Conclusão e Perspectivas Futuras

A revisão de código tem sido uma ferramenta extremamente útil para elevar o índice de qualidade do software, propiciando aprendizado contínuo e possibilitando que programadores mais experientes passem para os mais novos as melhores práticas de programação.

Notamos que é criado um clima de cumplicidade entre os integrantes do time, pois se algum problema ocorreu no cliente e não foi detectado no processo de revisão ou teste, a responsabilidade é de toda a equipe (que aprovou as modificações através da revisão) e não apenas do programador.

Devemos ter em mente que o processo de revisão não é a solução para todos os problemas e sim uma das muitas ferramentas ou procedimentos que visam a melhoria da qualidade nos produtos de software.

Para o futuro, podemos sugerir a criação de uma ferramenta de análise dos resultados que possa sugerir modificações no processo e identifique pontos de melhoria a fim de reduzir ainda mais o número de defeitos do software. Esta ferramenta analisaria todos os processos de revisão de vários projetos, permitindo assim a transferência de aprendizado entre projetos.

Referências Bibliográficas

McConnel, Steve (1996) "Rapid Development: Taming Wild Software Schedules", USA, Microsoft Press, p. 447.

Humphrey, Watts S. (1989) "Managing the Software Process", Addison-Wesley, p. 171-189.