Boas Práticas de Manutenção de Software

Fabrício Melo Ferreira, Tiago Netzel

Faculdades Integradas Camões
80020-040 - Curitiba – PR – Brazil

Curso Superior em Análise e Desenvolvimento de Sistemas

Abstract. *This article seeks to clarify this process that is little commented, but that is very important for the life of a software, we are talking about software maintenance, the life of a software does not end after it is implemented or developed, to last many years should be maintained in order to improve and increase its useful life and make life easier for users.*

Resumo. *Este artigo busca esclarecer esse processo que é pouco comentado, mas que é muito importante para vida útil de um software, estamos falando da manutenção de software, a vida de um software não acaba depois que é implementado ou desenvolvido, para que dure muitos anos deve ser feita a manutenção a fim de melhorar e aumentar sua vida útil e facilita a vida dos usuários.*

1. Introdução

Neste artigo, veremos o que é, como funciona e o e as etapas a serem realizadas para a manutenção de um software, o processo de otimização e melhoria de um software já produzido ou em fase de desenvolvimento. Na manutenção será possível corrigir erros percebidos durante a utilização pelo usuário ou testes fitos por desenvolvedores. O ciclo de vida de um software, não acaba após sua implementação no ambiente do usuário, pode durar muitos anos, agregando atualizações e ganhando novas versões.

A manutenção de um software se difere e muito da manutenção de produtos industrializados por exemplo, a manutenção de software não é realizada para reparar danos e defeitos causados pelo tempo de uso. Os defeitos e falhas não são resultados de longo tempo de uso e nem carga de utilização, mas sim problemas já existentes desde a fase de desenvolvimento que não foram detectados em nenhuma fase que passou. Atualmente existem três tipos de manutenção de software: Adaptativas, Corretivas e Evolutivas:

Adaptativas: são modificações que buscam adaptar o software a uma nova experiência ou novo ambiente externo, na maioria das vezes obrigatória. Como por exemplo: regras de negócio, mudanças na constituição, leis ou exigências do governo e órgãos reguladores. Assim como o desenvolvimento de uma plataforma de hardware, como processadores mais modernos e periféricos onde o software possa aproveitar esses recursos. E uma versão nova de um SO que talvez não seja totalmente compatível ao novo software.

Corretivas: são manutenções para resolver as falhas descobertas durante a produção, as corretivas são comuns em casos de baixo nível de preocupação com a qualidade do processo de desenvolvimento. Assim, buscam solucionar defeitos de funcionalidade e de uso.

Evolutivas: alterações não previstas no documento original do software. Servem para adicionar novas funcionalidades e aplicações solicitadas pelos usuários, melhorar seu desempenho e soluções que trazem mais facilidade para o usuário.

Ainda neste artigo, veremos mais algumas definições de manutenção de software e as dificuldades encontradas pelo caminho.

.

2. Manutenção de software

A manutenção de software e uma das áreas que abrangem a engenharia de software,ela e uma das fase que ocorre no processo de desenvolvimento de software, que tem como sua definição pela [IEEE, Institute], como a modificação de um produto de software depois de sua entrega (ao cliente) para corrigir erros, melhorar sua performance ou qualquer outro atributo, ou para adaptar o produto o produto a um ambiente modificado.

A fase de manutenção também esta ligada direto ao processo de desenvolvimento do modelo cascata, sendo a ultima etapa.

O processo de manutenção visa, logo após o software ser liberado para o usuário, modificar o produto para a correção de falhas, melhoria no desempenho ou outros atributos ou adaptar o sistema a uma mudança de ambiente. Cada pedido de mudança dependente do processo a ser realizado pra o atendimento do pedido entra em uma das categorias de manutenção existente, com base em alguns estudos, montou-se um gráfico que apresenta a porcentagem das ocorrências dos tipos de manutenção:

2.1 Manutenção corretiva

É a manutenção onde ocorrem os reparos dos erros encontrados durante a fase de teste do desenvolvimento, em determinadas situações esses erros podem não atrapalhar o funcionamento do software e não leva a problemas maiores, já que é possível resolve-los por meios simples, mas pode haver casos onde é necessário um reparo temporário para o funcionamento do software para solucionar algum problema, com o objetivo de colocar o sistema operando sua parte básica novamente, assim conseguindo tempo para reparar o problema e lançar o software operando todas as suas funções em uma nova versão.

2.2 Manutenção preventiva

É a alteração no software visando buscar uma melhor confiabilidade ou o oferecimento de uma melhor estrutura para manutenções futuras. Ela parte de uma observação feita pelos mantedores do sistema sobre o que poderá gerar algum tipo erro no software, de tal forma que esse erro será corrigido antes mesmo que ele possa vir a ocorrer.

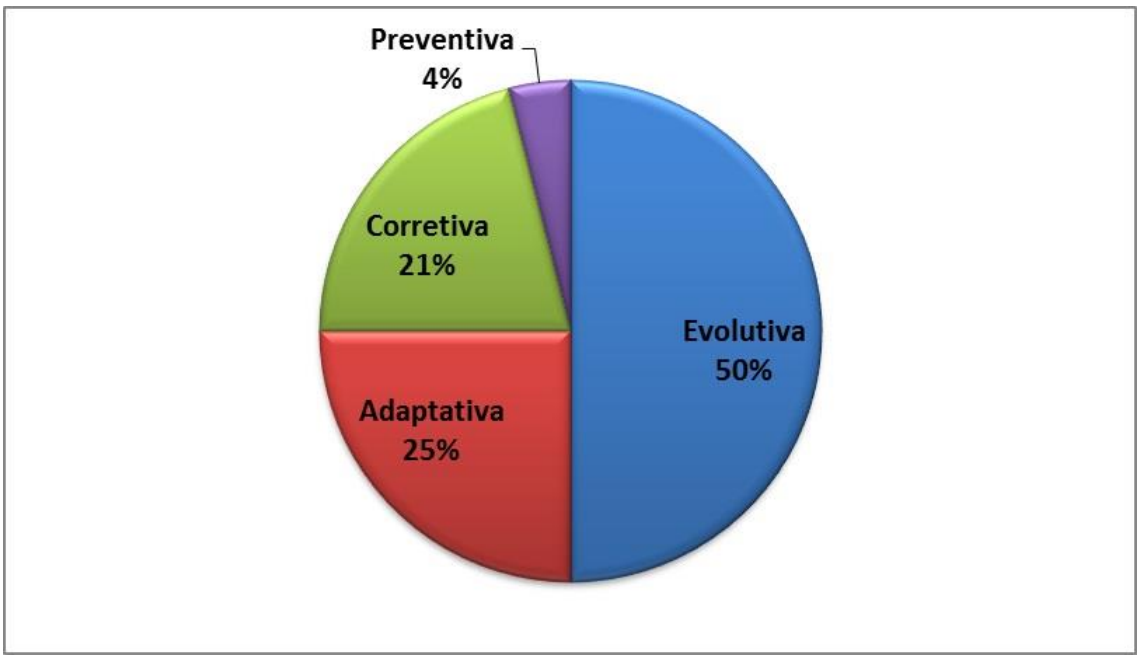
2.3 Manutenção adaptativa

É a adaptação do sistema para acomodar ele as constantes mudanças ocorridas em seu ambiente externo, essas mudanças podem ser regras de negócios, constituição e leis que venham a alterar a funcionalidade do software, podendo ser também a

adaptação do sistema a novas plataformas de hardwares ou até mesmo uma nova versão de um sistema operacional.

2.4 Manutenção evolutiva

É a manutenção que não estava previsto no escopo do projeto quando o software começou a ser desenvolvido, tem como objetivo deixar o sistema cada vez melhor, com o acréscimo de novas funcionalidades, melhoramento de desempenho, ou na modificação de seu código fonte na busca de melhor legibilidade ou adequação a açguns paradigmas de programação.



3. Prática

Melhores práticas para a manutenção de software inclui um estudo sobre a situação atual do software, a coleta de feedbacks dos usuários finais, mantendo-se maleável e definição de padrões de controle de qualidade.

Quanto mais preparado for o departamento de TI quando avalia a situação atual, mais fácil será pra traçar um caminho a manutenção de software. O departamento de TI deve esclarecer os problemas e as necessidades da empresa, analisar o que o software aplicativo atual faz contra o que ele precisar fazer, mas não faz. Nessa etapa do processo, inclui fazer estimativas de custo, criar potenciais cronogramas e analisar toda documentação original do aplicativo de software, incluindo uma cópia do código-fonte. A equipe responsável vai definir se o aplicativo só precisa ter algumas partes do código atualizado, se é necessário reorganizar e reestruturar os bancos de dados atuais ou refazê-lo desde o início usando como referência o código original. O departamento de TI deve receber e examinar todos os feedbacks das pessoas que utilizam as aplicações do software diariamente, Esses usuários finais, tem a chave do q o software precisa fazer para ser mais eficiente, mais seguro e eficaz. Como parte das melhores práticas, o departamento de TI deve resumir o feedback recebido em um documento de no máximo duas páginas e disponibilizá-las aos usuários finais para que fique informados sobre o progresso da atualização.

É esperado que tudo saia de acordo com o planejamento preliminar do departamento de TI a situação atual. Porem, é necessário manter-se completamente flexível e ter o pensamento de que os planos e as estimativas preliminares pode e vai mudar toda a linha de progresso durante o processo de manutenção. Acontecimentos inesperados como pessoal doente, restrições orçamentárias, variações políticas e mais podem afetar prazos e custos do projeto. Para se preparar para esse eventuais situações, a equipe responsável deve formar um ou mais planos de contingência para manter os progressos de manutenção progredindo.

Qualidade garantia de informação deve ser definido antes do inicio da manutenção atual. Determinar pontos de referência e como os objetivos serão testados. Uma boa idéia para acompanhar o processo de manutenção para a eficiência e revê-lo em conformidade, sempre dando ênfase para data de conclusão. Essa etapa do processo inclui definir o que deve ser documentado em todo o processo de manutenção, como a nova documentação vai interagir com a antiga, e como o aplicativo de software pode ser mantido no futuro. A melhor prática é perceber que testes exaustivos nunca são a melhor solução. Interação do usuário final vai encontrar bugs e problemas que de outra forma não seriam descobertos.

.

References

- **PRESSMAN, Roger S**. Engenharia de software. 6ª.ed. São Paulo : McGraw-Hill, 2006. xxxi, 720p.
- **BENNETT, K. H.; RAJLICH, V. T. (2000)** "SOFTWARE MAINTENANCE AND EVOLUTION: A ROADMAP", IN: CONFERENCE ON THE FUTURE OF SOFTWARE ENGINEERING, LIMERICK, IRELAND, JUNE.
- **BENNETT, K. H.; RAMAGE, M.; MUNRO, M. (1999)** "DECISION MODEL FOR LEGACY SYSTEMS", IEEE PROCEEDINGS ON SOFTWARE (TSE), V.146, N. 3, P. 153-159.
- **BHATT, P.; SHROFF, G.; MISRA, A. K. (2004)** "DYNAMICS OF SOFTWARE MAINTENANCE", ACM SIGSOFT SOFTWARE ENGINEERING NOTES, V. 29, N. 5, P. 1-5.
- **BASILI, V. (1990)** "VIEWING MAINTENANCE AS REUSE-ORIENTED SOFTWARE DEVELOPMENT", IEEE SOFTWARE, V. 7, N. 1, P. 19-25.
- **CHAPIN, N. (1986)** "SOFTWARE MAINTENANCE: A DIFFERENT VIEW", IN: CONFERENCE ON SOFTWARE MAINTENANCE, ORLANDO, FL, USA, NOVEMBER.
- **LIENTZ, B.P; Swanson, E.B. (1980)** "Software Maintenance Management", Reading, MA, Addison Wesley.
- **GHEZZI, C.; MANDRIOLI, D. (2005)** "The Challenges of Software Engineering Education", In: 27th International Conference on Software Engineering, Saint-Louis, Missouri, USA, May.
- **HAZZAN, O.; DUBINSKY, Y. (2003)** "Teaching a Software Development Methodology: The case of Extreme Programming", In: 16th Conference on

Software Engineering Education and Training (CSEE&T 2003), Madrid, Spain, March.

- **IEEE (1998)** "Std 1219 – IEEE Standard for Software Maintenance", Institute of Electrical and Eletronic Engineers, New York, NY, USA.