ENGENHARIA DE SOFTWARE

Izabelly Soares de Morais

SAGAH
SOLUÇÕES EDUCACIONAIS INTEGRADAS

Manutenção de software

Objetivos de aprendizagem

Ao final deste texto, você deve apresentar os seguintes aprendizados:

- Explicar os conceitos gerais de manutenção de software.
- Identificar os problemas que podem ocorrer na manutenção de software.
- Analisar os motivos de mudanças de software.

Introdução

A palavra "manutenção" é originada do latim *mantenus tenere*, que significa "manter aquilo que se tem". Na engenharia de software, a manutenção do software é a última etapa do seu desenvolvimento, quando ele já está pronto, operacional, e precisa de reparos e modificações para continuar atendendo novos requisitos que surgem. Um software criado e que não possui manutenção pode ser pouco utilizado ou até mesmo inutilizado. Constantes atualizações para corrigir falhas e realizar melhorias são essenciais para que ele continue "vivo".

Neste capítulo, você vai adquirir conhecimentos fundamentais para entender o processo de manutenção de software e os problemas que podem ocorrer na sua manutenção. Você vai ver, também, alguns dos motivos de mudanças e a necessidade da manutenção.

Manutenção de software: conceitos gerais

O desenvolvimento de software não é interrompido quando o sistema é entregue, mas continua por toda a vida útil do sistema. Depois que o sistema é implantado, para que ele se mantenha útil é inevitável que ocorram mudanças. A evolução do software é importante, pois as organizações investem grandes quantias de dinheiro em software e são totalmente dependentes desses sistemas. Seus sistemas são ativos críticos de negócios, e as organizações devem investir nas mudanças de sistemas para manter o valor desses ativos.

Consequentemente, a maioria das grandes empresas gasta mais na manutenção de sistemas existentes do que no desenvolvimento de novos sistemas (SOMMERVILLE, 2011).

Todo sistema é desenvolvido para uma finalidade, ou seja, resolvem algum problema ou ajudam os usuários em alguma atividade específica. Porém, estamos sempre em constante processo de mudanças, nas quais, com o passar do tempo, algumas coisas acabam obsoletas. O mesmo ocorre com um software, ou seja, não quer dizer que, porque um software foi desenvolvido com todas as funcionalidades operando corretamente para solucionar algum problema, ele nunca precisará de alguma manutenção.

O ciclo de vida de um software é composto por algumas fases: definição de requisitos, projeto de software, implementação e teste de unidades, integração e teste do sistema e, por fim, operação e manutenção do software. Dessa forma, não importa a finalidade do software, em algum momento este irá demonstrar sinais de que precisa passar por uma manutenção, seja por meio da atualização de suas funções, ou até mesmo para que configurações do sistema sejam adaptadas aos requisitos do sistema.

A manutenção de um software, como o próprio termo já diz, promoverá a realização de mudanças no sistema, ou até mesmo a adaptação deste com um novo contexto, no qual ele pode ser inserido. Sabemos que tudo precisa de manutenção e com o software não é diferente.

Fique atento

É importante salientar que toda modificação em um software irá interferir na questão do custo-benefício. Dessa forma, mesmo que mudanças aconteçam no software, mesmo quando ele já tenha sido entregue ao usuário, irá ter algum custo financeiro. Nem sempre as mudanças são cabíveis; muitas vezes, é mais barato desenvolver outro software do que inserir diversas mudanças em algum já existente. Cabe à equipe de desenvolvimento ou de manutenção avaliar, pois pode variar de software para software.

Motivos das mudanças de software

Para Sommerville (2011), prever o número de solicitações de mudança para um sistema requer uma compreensão do relacionamento entre o sistema e seu ambiente externo. Alguns sistemas têm relacionamento muito complexo com seu ambiente externo, e as mudanças nesse ambiente inevitavelmente resultam em alterações. Sob o ponto de vista de Sommerville (2011), historicamente sempre houve uma separação entre o processo de desenvolvimento e o de manutenção do software. Os custos da manutenção geralmente são mais altos do que os custos iniciais do desenvolvimento. Os processos de manutenção são, em alguns casos, considerados menos desafiadores do que o desenvolvimento do software original. A distinção entre o desenvolvimento e a manutenção do software é vista muitas vezes como irrelevante. Poucos sistemas de software são completamente novos, e faz muito mais sentido ver o desenvolvimento e a manutenção como processos contínuos. A Figura 1 a seguir mostra algumas etapas que podem ser seguidas.

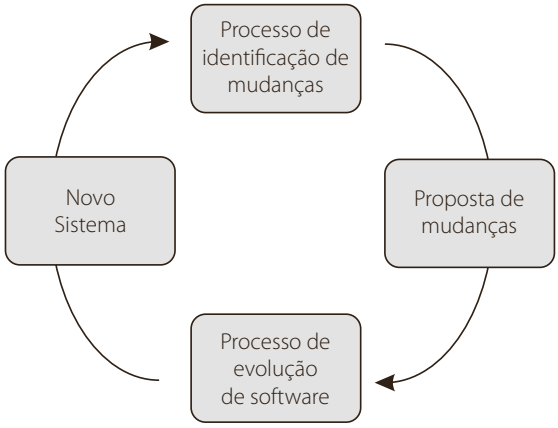

Figura 1. Etapas para a manutenção.

Fonte: Sommerville (2011, p. 166).

Dessa forma, podemos perceber que a manutenção do software pode mesmo ser vista como uma evolução, na qual, de acordo com Sommerville (2011), existe uma fase em que há a identificação de mudanças e, posteriormente, as propostas devem ser avaliadas, pois, como já comentamos neste capítulo, as mudanças sempre interferem na relação de custo-benefício, por isso há o processo de evolução do software, no qual esse processo pode ser viável, ou não.

De acordo com Schach (2010), existem três razões principais para realizar modificações em um produto:

1. Uma falha precisa ser corrigida, seja ela uma falha de análise, projeto, decodificação, documentação ou qualquer outro tipo. Isso é denominado **manutenção corretiva**. Neste ponto, Sommerville (2011) acrescenta que a falha, ou defeito, geralmente é grave o suficiente para impedir a continuidade do funcionamento normal do sistema.
2. Na manutenção de aperfeiçoamento, é feita uma modificação no código para aumentar a eficácia do produto. Por exemplo, o cliente pode querer funcionalidade adicional ou solicitar que o sistema seja modificado para que fique mais rápido. Aumentar a facilidade de manutenção de um produto é outro exemplo de manutenção de aperfeiçoamento.
3. Na manutenção adaptativa, é feita uma modificação no produto como resposta a uma modificação no ambiente em que ele opera. Por exemplo, é quase certo que um produto tenha de ser modificado se ele for portado para um novo compilador, um sistema operacional ou um hardware. A cada mudança na legislação tributária, um produto que ajuda na preparação de declarações de imposto de renda precisa ser modificado de acordo. Quando o correio norte-americano introduziu os CEPs com nove dígitos em 1981, os produtos que permitiam o uso de CEPs com apenas cinco dígitos tiveram de ser modificados. A manutenção adaptativa não é solicitada pelo cliente; pelo contrário, ela é imposta ao cliente por um agente externo.

Para Sommerville (2011), fica claro que os programadores de manutenção devem ter praticamente todas as habilidades técnicas que um profissional de software deve ter. Mas o que ele ganha em troca?

- A manutenção pós-entrega é um trabalho ingrato em todos os aspectos. Os profissionais de manutenção têm de lidar com usuários insatisfeitos, pois, se estivessem contentes com o produto, este não precisaria de manutenção.

- Os problemas do usuário frequentemente são causados por aqueles que desenvolveram o produto, e não pelo profissional de manutenção.
- O próprio código poderia ter sido malfeito, o que é mais um fator que se acumula às frustrações do profissional de manutenção.
- A manutenção pós-entrega é menosprezada por muitos desenvolvedores de software, que consideram a atividade de desenvolvimento de software um trabalho glamouroso e a manutenção, um trabalho escravo, apropriado apenas para programadores novatos ou incompetentes. Ela pode ser comparada ao atendimento pós-venda. O produto foi entregue ao cliente, porém, ele está insatisfeito, pois o produto não funciona corretamente, não faz tudo o que deseja ou as circunstâncias para as quais ele foi construído mudaram de alguma forma. Existem diversos problemas que podem ser encontrados nessa fase, os quais veremos no próximo tópico.

Problemas que podem ocorrer na manutenção de software

Conforme Pressman e Maxim (2016), enfrentamos uma fila cada vez maior de correções de erros, solicitações de adaptação e melhorias que devem ser planejadas, programadas e, por fim, executadas. Logo, a fila já cresceu muito e o trabalho ameaça devorar os recursos disponíveis. Com o passar do tempo, sua empresa descobre que está gastando mais tempo e dinheiro com a manutenção dos programas do que criando novas aplicações. Não é raro uma empresa de software despender de 60% a 70% de todos os recursos com manutenção de software.

Pressman e Maxim (2016) questionam sobre a necessidade da realização de tanta manutenção e se tanto esforço deve ser despendido. Traz, ainda, uma possível resposta dada por Osborne e Chikofsky (1990 apud PRESSMAN; MAXIM, 2016, p. 797): “Muitos softwares dos quais dependemos hoje têm de 10 a 15 anos, em média. Mesmo quando esses programas foram criados, usando as melhores técnicas de projeto e codificação conhecidas na época (e muitos não foram), o tamanho do programa e o espaço de armazenamento eram as preocupações principais. Eles então migraram para novas plataformas, foram ajustados para mudanças nas máquinas e na tecnologia dos sistemas operacionais e aperfeiçoados para atender a novas necessidades dos usuários – tudo isso sem grande atenção à arquitetura geral. O resultado são estruturas

mal projetadas, mal codificadas, de lógica deficiente e mal documentadas em relação aos sistemas de software, e que devemos manter funcionando [...]".

Outra razão para o problema de manutenção do software é a mobilidade dos profissionais. É provável que a equipe (ou pessoa) responsável pelo trabalho original não esteja mais por perto, ou, pior ainda, que outras gerações de profissionais de software possam ter modificado o sistema e já não estejam mais presentes. Hoje, pode não ter restado ninguém que tenha conhecimento direto do sistema legado (PRESSMAN; MAXIM, 2016). Se pararmos para analisar, a manutenção do software pode trazer o problema de métricas escolhidas durante seu processo de produção, ou seja, o sistema final é fruto de diversas escolhas que abrangem as mais variadas vertentes de um software. Dessa forma, isso pode ser considerado mais um problema para a realização da manutenção pós-entrega de um software.

Nesta etapa, podemos notar a importância da documentação do software e o modo como essa documentação foi concebida. Sua eficácia é ratificada neste momento, no qual, para realizar a manutenção, a equipe terá como base a documentação produzida, a qual deve trazer os mínimos detalhes sobre o software.

De acordo com Schach (2010), um relatório de defeitos será preenchido, se, na opinião do usuário, o produto não estiver funcionando como especificado no manual do usuário. Existem várias causas possíveis. Primeiro, pode ser que não haja nada de errado; pode ser que o usuário tenha interpretado mal o manual ou esteja usando o produto incorretamente. De modo alternativo, se existe uma falha no produto, pode ser que simplesmente o manual do usuário tenha sido mal redigido e não haja nada de errado com o código. Entretanto, normalmente existe alguma imperfeição no código. Porém, antes de fazer qualquer modificação, o programador de manutenção deve determinar exatamente onde está a imperfeição e usar o relatório de defeitos preenchido pelo usuário, o código-fonte, nada mais. Consequentemente, o programador de manutenção precisa ter habilidades de depuração muito acima da média, pois a imperfeição poderia estar em qualquer ponto do produto. A causa original do defeito poderia estar, até o momento, nos artefatos de projeto ou de análise.

Conclui-se que o processo de desenvolvimento de software deve ser percorrido com muita cautela e a escolha de métodos, ferramentas e equipe poderá afetar não só o processo, mas também o resultado final, que é o sistema, e, consequentemente, a sua usabilidade estará atrelada a todos esses fatores citados anteriormente. Dessa forma, é sempre importante a realização de uma

preparação inicial antes do começo de ciclo de vida do software. A manutenção requer atenção tanto quanto as demais etapas do processo, pois ela garante o tempo de vida útil do software.

Saiba mais

Nesta fase de manutenção de software, um documento específico pode ser desenvolvido, chamado de **relatório de defeitos**.

A primeira coisa necessária ao se fazer a manutenção de um produto é um procedimento para modificá-lo. Em relação à manutenção corretiva, isto é, à eliminação de falhas residuais, se o produto não estiver funcionando corretamente, então o usuário deverá preencher um relatório de defeitos, que deve conter informações suficientes para permitir que o programador de manutenção repare o problema. Normalmente trata-se de algum tipo de defeito de software. Além disso, o programador de manutenção deve indicar a gravidade do defeito. Categorias de gravidade típicas são: crítico, importante, normal, secundário e trivial (SCHACH, 2010).

Referências

OSBORNE, W. M.; CHIKOFSKY, E. J. Fitting pieces to the maintenance puzzle. *IEEE Software*, Los Angeles, p. 11-12, Jan. 1990.

PRESSMAN, R. S.; MAXIM, B. R. *Engenharia de software*: uma abordagem profissional. 8. ed. Porto Alegre: AMGH, 2016.

SCHACH, S. R. *Engenharia de software*: os paradigmas clássicos e orientado a objetos. 7. ed. Porto Alegre: AMGH, 2010.

SOMMERVILLE, I. *Engenharia de software*. 9. ed. São Paulo: Pearson, 2011.

Leitura recomendada

PFLEEGER, S. L. *Engenharia de software*: teoria e prática. 2. ed. São Paulo: Prentice-Hall, 2004.

Encerra aqui o trecho do livro disponibilizado para esta Unidade de Aprendizagem. Na Biblioteca Virtual da Instituição, você encontra a obra na íntegra.

Conteúdo:

SAGAH
SOLUÇÕES EDUCACIONAIS INTEGRADAS