SESI SENAI ITAJUBÁ CFP AURELIANO CHAVES

ADRIANO FLAVIO MELO
DOUGLAS EVANDO RIBEIRO BRAGA
JÉSSICA AMANDA RIBEIRO BRAGA
NÚBIA THAYNARA

MANUTENÇÃO DE SISTEMAS

ITAJUBÁ
2023

Sumário

Manutenção de Software

Definição

Quando se trata de desenvolvimento de software, sabemos que não existe um fim definitivo. Seja por necessidades tecnológicas ou de negócios, a manutenção de software é uma atividade que acompanhará a vida toda de um sistema. Atualmente, as empresas investem muito em TI, e a maior parte desse recurso é destinado para a evolução e manutenção de software

Isso porque os sistemas são fundamentais para que os negócios operem, e para garantir seu funcionamento adequado e evolução conforme as necessidades da empresa, é necessária uma equipe qualificada dedicada a essa função.

Manutenção de software é o processo de melhorias e correções de um software em desenvolvimento ou já desenvolvido. Ou seja, qualquer alteração após o software estar disponível em produção.

A vida do software não termina após sua implantação. Para um software permanecer útil ao longo do tempo, é necessário investir em manutenção.

Um dos principais objetivos em realizar a manutenção de software é o aperfeiçoamento tecnológico do sistema. Ao realizar essa atividade, o sistema se tornará cada vez mais estável, diminuindo sua velocidade de envelhecimento.

Os negócios estão sempre em mudanças, e o software deve acompanhar esse movimento. Dessa forma, é fundamental que o produto se adapte a novas regras de negócios, e acumule novas funcionalidades de acordo com as necessidades das partes envolvidas.

Técnicas para manutenção de software

Muitas ferramentas e técnicas auxiliam no processo de manutenção de software, evitando erros gerais durante o desenvolvimento, sustentação e evolução do sistema.

Documentação

Muitas vezes negligenciada, a documentação tem grande importância não só no momento de criação do software, mas também durante sua manutenção. Um dos

motivos para manter uma boa documentação de um software está relacionado a mudanças de equipes de desenvolvimento. Com a documentação, o processo de transferência de conhecimento sobre o sistema se torna mais eficaz, não dependendo de pessoas especificas que já passaram pelo projeto.

Versionamento

Toda vez que há um processo de build e deploy, uma nova versão de um software é disponibilizada. O versionamento de uma aplicação tem como objetivo documentar as alterações feitas em cada entrega, de modo que seja possível resgatar versões anteriores em caso de erros.

Status Reporting

Status Reporting é um documento que visa o alinhamento entre os envolvidos no projeto em relação a situação atual do desenvolvimento. Nele é registrado o que foi realizado pela equipe de desenvolvimento em determinado período, assim como pontos de atenção relacionados ao sistema.

Codificação

Boas práticas de codificação ajudam muito na legibilidade, e auxiliam também para que o ciclo de evolução seja mais ágil e fácil. Endentação e comentários no código podem parecer detalhes não muito relevantes, mas irão auxiliar muito no entendimento do código-fonte. O uso de pacotes, orientação a objetos e padrões de projeto também contribuem na organização e divisão das responsabilidades para reaproveitamento de código, aumentando a produtividade das equipes de desenvolvimento.

Custos e Desafios

Alguns problemas podem ocorrer durante a realização de algum processo de manutenção. Eles devem ser na medida do possível previstos para poder ser evitados, tais como:

- Ausência ou deficiência na documentação;
- Dificuldade na identificação de manutenções realizadas anteriormente;
- Falta de controle de versão;
- Baixa manutenibilidade do software.
- Ausência de um processo de manutenção de software;

- Grande expectativa dos usuários;
- Elevada rotatividade de membros e funções dentro da equipe;
- Sobrecarga de tarefas;
- Estimativa de prazo não condizente com a complexidade do software;
- Baixa motivação entre profissionais de manutenção;
- Ausência de manutenção preventiva;
- Falhas de comunicação com o usuário;
- Atrasos na entrega.

Além disso, efeitos colaterais podem ocorrer com a realização desses processos:

- Modificação da estrutura do código;
- Inclusão de códigos com erros;
- Desatualização da documentação.

Através dessas características, deve-se identificar e registrar todas as partes que foram afetadas pela modificação.

Um dos principais desafios se refere à questão financeira, afinal recurso financeiro normalmente é escasso e não pode ser jogado fora. Talvez seja o recurso mais crítico para o desenvolvimento de software e um levantamento de requisitos bem executado contribui muito para que se tenham custos em níveis razoáveis.

Outro desafio importante é o tempo. Com prazos cada vez menores e os problemas mais difíceis muitas, vezes fases importantes não são executadas satisfatoriamente. Testes deixam de ser realizados gerando um produto não tão confiável ou estável. Isso pode gerar o retrabalho e em consequência mais tempo para desenvolver.

Um software estável, confiável, robusto pode ser tudo que o cliente deseja, mas também deve ser de fácil aprendizado. Um software deve resolver o que se propõe na sua concepção e ao mesmo tempo deve ser de manipulação agradável por parte do cliente. Este é um equilíbrio que deve ser alcançado e a área de usabilidade pode dar preciosas contribuições para isto.

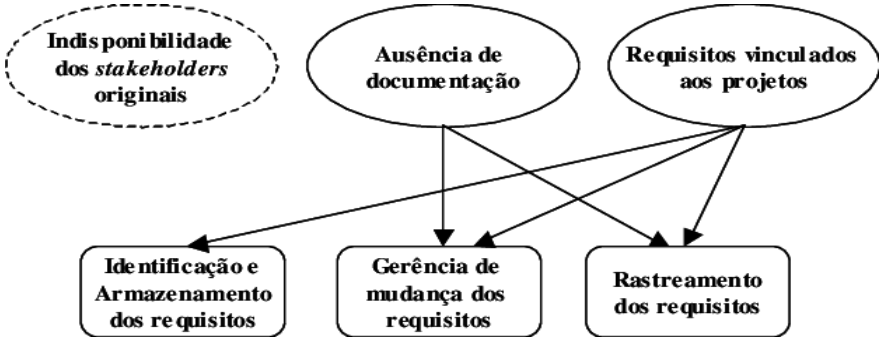
Para a manutenção de software, os sistemas devem sofrer mudanças de acordo com o contexto em que estão inseridos. Para melhorar o processo de manutenção devem ser levadas em conta as mudanças ocorridas no ambiente em que está inserido o sistema.

Na manutenção de software existem quatro fases que são introdução, crescimento, maturidade e declínio. Destaca-se que durante a fase de introdução existe um grande suporte ao usuário, período em que são concebidas as primeiras ideias sobre o sistema. As fases de maturidade e de crescimento do software são de ajustes, onde o sistema já está concretizado e muitas vezes correções aparecem naturalmente. Ainda na fase de maturidade, podem ser realizadas melhorias no programa. Durante essas fases é necessário que sejam realizados testes e atualização da documentação para avaliar as partes modificadas e acrescentadas na aplicação. Na fase de declínio o sistema chega ao final da vida útil e deve ser avaliado se o sistema deve ser retirado de operação. A avaliação deve ser feita com base nos aspectos econômicos como, por exemplo, substituição de tecnologias, ou até mesmo para conseguir independência de fabricante.

A razão do custo elevado deve-se, em parte, à própria natureza da atividade de manutenção, caracterizada principalmente pela imprevisibilidade. Além dos altos custos financeiros, essa é também a atividade que exige maior esforço dentre as atividades de engenharia de software.

A manutenção de software é uma operação importante, pois consome a maior parte dos custos envolvidos no ciclo de vida de um software, e a falta de habilidade em mudar um software rapidamente, e de maneira confiável, pode causar a perda de oportunidades de negócio.

Embora não exista um consenso sobre o valor exato do custo atrelado à atividade de manutenção, as pesquisas na área apontam, na totalidade dos casos, sempre mais de 50% dos investimentos realizados no software. A importância financeira atrelada à manutenção de software é ainda agravada quando se leva em consideração o risco para as oportunidades de negócio, que podem ser causadas pela falta de gerenciamento e compreensão total da dinâmica da atividade. Esse gerenciamento deve considerar três fatores: ferramentas, pessoas, processos, revelando-se, pois, uma atividade gerencial.



Efeitos colaterais

Quando se altera o software, seja para correção, evolução ou adaptação, existe a possibilidade de se introduzir novos defeitos ou reintroduzir erros que ocorriam anteriormente no mesmo. É o chamado efeito colateral. O ideal seria que todos os módulos ligados à parte alterada ou o sistema todo fossem novamente testados. Mas devido aos curtos prazos, principalmente em manutenções corretivas emergenciais, quase nunca isso é possível. A falha causada através do efeito colateral pode aparecer em qualquer ponto do software, podendo ficar fora da cobertura dos testes daquela manutenção.

Isto é um desafio para os testadores, pois eles precisam garantir o correto funcionamento do sistema como um todo. Uma forma de prevenir este efeito seria a realização dos testes de regressão, que são realizados para garantir que os novos defeitos introduzidos ou desmascarados sejam encontrados e removidos.

Para atender aos curtos prazos, os testes de regressão devem ser automatizados. Realizá-los manualmente não seria viável. Entretanto, automatizar é uma tarefa complexa e custosa, que exige grande esforço. Mas, uma vez automatizado, o teste de regressão torna-se muito mais econômico, correspondendo apenas a uma fração do tempo gasto caso fosse realizado manualmente. Uma sugestão para diluir o alto custo da automação dos testes de regressão, seria a automação de partes do sistema a cada nova demanda de manutenção, alcançando a situação ideal no médio ou longo prazo.

A automação é particularmente importante para obter um nível razoável de garantia em um ciclo de desenvolvimento muito rápido, como por exemplo, o ciclo das manutenções corretivas. Uma vez atingido esse cenário ideal, os testes serão rápidos e eficazes, mitigando os riscos das falhas causadas pelo efeito colateral.

Custos e Desafios

A manutenção de software é uma operação importante, pois consome a maior parte dos custos envolvidos no ciclo de vida de um software, e a falta de habilidade em mudar um software rapidamente, e de maneira confiável, pode causar a perda de oportunidades de negócio.

Embora não exista um consenso sobre o valor exato do custo atrelado à atividade de manutenção, as pesquisas na área apontam, na totalidade dos casos, sempre mais de 50% dos investimentos realizados no software. A razão do custo elevado deve-se, em parte, à própria natureza da atividade de manutenção, caracterizada principalmente pela imprevisibilidade. Além dos altos custos financeiros, essa é também a atividade que exige maior esforço dentre as atividades de engenharia de software. Pressman (2005) ainda completa que o grande esforço necessário na manutenção se justifica pela abrangência do significado desse termo no contexto de software.

A importância financeira atrelada à manutenção de software é ainda agravada quando se leva em consideração o risco para as oportunidades de negócio, que podem ser causadas pela falta de gerenciamento e compreensão total da dinâmica da atividade. Esse gerenciamento deve considerar três fatores: ferramentas, pessoas, processos, revelando-se, pois, uma atividade gerencial complexa.

Se por um lado a atividade de manutenção é dispendiosa, por outro, ela é um desafio para as organizações que precisam considerá-la em seu dia-a-dia. Não é de se esperar que uma empresa de grande porte troque todos seus sistemas somente pelo fato de que a tecnologia neles empregada está ultrapassada. Esses sistemas representam ativos importantes da organização e ela estará disposta a investir de maneira a manter seus valores.

Prever quando uma manutenção precisará ser conduzida é uma tarefa geralmente muito difícil de ser realizada. Essa dificuldade de previsão, e também controle sobre a manutenção, é explicada pelo fato de muitas vezes ficar a cargo de empresas terceirizadas a manutenção, revelando-se um problema já que normalmente essas organizações não possuem nenhum contato com o projeto inicial do software. O aumento na complexidade dos softwares produzidos (tanto em termos de funcionalidades, como de técnicas) torna a previsão de esforços de manutenção muito vaga. Essa dificuldade em estimar esforços torna-se mais evidente quando se trata de sistemas legados.

Ferramentas de Apoio

Hoje no mercado existem ferramentas que foram desenvolvidas para auxiliar os desenvolvedores a realizarem a manutenção de software.

Atualmente existem ferramentas que possuem recursos direcionados ao gerenciamento de equipamentos e auxílio à mão-de-obra, por exemplo, tentando maximizar tempo e produtividade. Podem também direcionar recursos para correção e prevenção de falhas nos produtos, além de gerar relatórios, gráficos e indicadores gerenciais. Esses relatórios e indicadores permitem que a gerência tenha um domínio maior sobre os processos de desenvolvimento e sobre os produtos construídos. Muitas dessas ferramentas permitem que o gerente do projeto receba notificações sobre algum ocorrido, como alarmes e anomalias.

Existem ainda outras ferramentas que podem ser usadas para auxiliar a equipe que irá realizar a manutenção do sistema. São ferramentas que realizam controle de versão das aplicações. O controle de versão permite o gerenciamento de diferentes versões do código fonte da aplicação ou da documentação do sistema. Com o versionamento, a manutenção de software pode se beneficiar de algumas funcionalidades que são muito úteis para implantação da manutenção no projeto, como o registro de todas as modificações realizadas ou a recuperação de uma versão específica de um produto.

Outra vantagem é que mais de um desenvolvedor pode realizar a manutenção em determinada parte do programa ao mesmo tempo. Ao alterar alguma parte do código, é possível unir diferentes modificações, e se o código realiza a mesma rotina, pode-se escolher a melhor implementação.

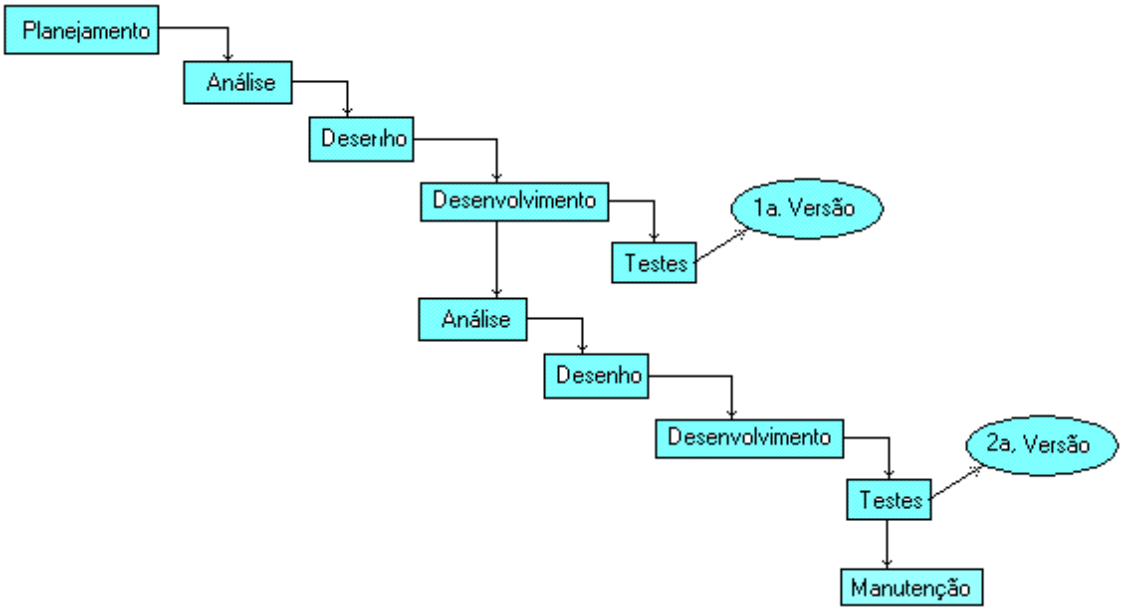
O que pode ser feito no código fonte pode ser feito nos documentos relacionados ao sistema. É primordial que a documentação esteja sincronizada e condizente com o que está sendo feito na aplicação. As ferramentas de controle de versão realizam o gerenciamento do código da aplicação, podendo, também, realizar o gerenciamento da documentação usada no sistema.

As ações ligadas à atividade de manutenção de software foram classificadas de acordo com sua natureza em três categorias: corretivas, adaptativas e perfectivas.

Processo de Manutenção

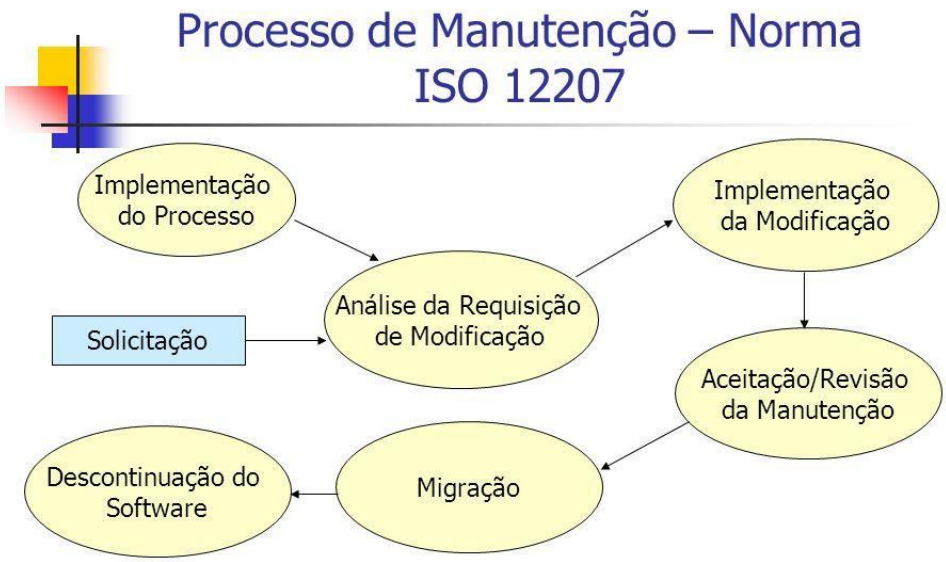
Um processo de Manutenção diz respeito a um conjunto de etapas bem definidas, que direcionam as atividades de manutenção de software, com o objetivo primordial de satisfazer às necessidades dos usuários de maneira planejada e controlada.

Segundo a norma NBR ISO 12207, o objetivo do processo de manutenção é modificar um produto de software existente, preservando a sua integridade. A Manutenção se inicia quando uma necessidade de modificação é identificada, ou seja, quando se necessita corrigir problemas, realizar adaptações ou melhorias. O processo de manutenção chega ao seu final no momento da descontinuação do software, ou seja, quando não se vai mais utilizá-lo.



Processo de Manutenção – Norma ISO 12207

- Implementação do Processo
- Implementação da Modificação
- Análise da Requisição de Modificação
- Solicitação
- Aceitação/Revisão da Manutenção
- Descontinuação do Software
- Migração



Implementação do Processo: nesta etapa, são estabelecidos planos e procedimentos para registrar e controlar a atividade de manutenção e os pedidos feitos pelos clientes.

Solicitação: este evento ocorre quando alguma solicitação de modificação é feita, ou pelos clientes, ou pelos próprios mantenedores.

Análise da Requisição de Modificação: nesta etapa, é feita uma verificação minuciosa da solicitação por parte do mantenedor, para que este possa oferecer opções de solução para o problema identificado.

Implementação da Modificação: nesta etapa, são realizadas as tarefas propriamente ditas de alteração do produto de software, incluindo código, documentação etc. Nela, deve-se garantir a perfeita execução para se chegar à solução proposta.

Aceitação/Revisão da Modificação: nesta etapa, são feitas as revisões e testes a fim de garantir a integridade do produto, bem como a homologação e aprovação junto ao solicitante para que o produto possa ser liberado.

Migração: nesta etapa, o produto gerado é colocado no ambiente de produção e uma avaliação deve ser conduzida para confirmar a execução perfeita da alteração.

Manutenção Adaptativa

A manutenção adaptativa visa a adaptação do software a uma nova regra de negócio, isto é, tem como objetivo adequar o sistema a um novo ambiente ou adequar melhor o software para que ele lide melhor no seu ambiente de funcionamento. Fornece as melhorias que um sistema precisa para acompanhar as mudanças e se adequar a necessidade dos usuários a cada alteração. Essas mudanças são o que mantém o sistema em dia e em pleno funcionamento, agradando os usuários o que é indispensável para manter as regras de negócio. Por exemplo, um sistema operacional precisa de mudanças ao sofrer um upgrade e algumas mudanças precisam ser feitas para acomodar um novo sistema operacional.

Um exemplo retirado do mundo real: antes da reforma trabalhista, um sistema de Recursos Humanos não permitia parcelar as férias em três vezes. Com a mudança da lei, isso passou a ser permitido, e o sistema teve que se adequar a esse cenário. Essas mudanças podem ser nas regras de negócio, mas podem vir de constituição e leis que tenham consequências na função do sistema, podendo ser uma nova geração de periféricos do qual o software se aproveita para melhor funcionamento, componentes do computador ou uma nova versão do sistema operacional.

É comum que mudanças adaptativas estejam correlacionadas a novas gerações de plataformas de hardware, sejam elas processadores ou memória RAM, essas mudanças adaptam o software a usar melhor ou usar mais da capacidade da máquina se adaptando para atender melhor quaisquer necessidades do seu usuário e usando o máximo possível do hardware utilizado pelo sistema para funcionar. Também pode ocorrer uma atualização para que o sistema continue funcionando, seja por compatibilidade com o sistema operacional ou qualquer outra. Um exemplo seria: atualizar um computador com sistema operacional de 32bits para 64bits para que ele siga conseguindo executar todos os programas disponíveis.

Importância

A importância da manutenção adaptativa reflete diretamente na relevância do aplicativo, se manter adaptado aos usuários e ao sistema no qual está inserido é essencial para um software se manter em uso.

Vantagens

Suas vantagens são refletidas diretamente ao usuário, sendo essas, um aumento na velocidade por utilizar mais da memória RAM ou uma nova função desejada pelos usuários ou desenvolvedores para facilitar ou auxiliar o uso de um software.

Dificuldades

Suas dificuldades são as mesmas que qualquer adaptação, sendo essas, acertar as necessidades e vontades do usuário no momento de se atualizar, é importante que as adaptações não desagradem o usuário ou afetem no funcionamento correto de um software.

Manutenção Corretiva

A manutenção corretiva vem para a correção de erros não encontrados na fase de teste, caso essa fase tenha existido, se não existir é mais provável que seja necessário a manutenção corretiva para corrigir os problemas que serão intensificados no software.

Em alguns casos os erros não impedirão o funcionamento do programa, podendo ser corrigidos com simples ajustes e manutenções rápidas que não necessitam de congelar o funcionamento do mesmo, já em alguns casos é preciso um reparo temporário que pode congelar o funcionamento do sistema para evitar mais erros ou para corrigir os erros já identificados, com isso podendo ou não deixar o sistema inutilizável por um período de tempo.

É comum que hajam problemas de funcionalidades, também conhecidos como "*bugs"* durante o desenvolvimento ou funcionamento de um sistema, afinal, mesmo que a equipe use a melhor das técnicas e etapas para desenvolver o software erros podem acontecer, por isso é tão necessário que todos programas que desejam se manter funcionais tenham manutenções corretivas, sendo essas, essenciais para a manutenção do software.

Vantagens

As vantagens da manutenção corretiva são visíveis no funcionamento do sistema já que corrigem um antigo problema, e para isso é preciso avaliar a documentação do projeto, a arquitetura, o impacto das modificações, a modificação no projeto original a

implementação de mudanças e testes de regressão, para garantir que as correções funcionarão corretamente quando lançadas ao sistema final.

Dificuldades

As desvantagens são possíveis impactos negativos, talvez um problema ou solução ocasione outros, mas esses também são necessários para a evolução do sistema implementado. Há também dificuldades em um possível congelamento das funções do software além dos impactos que essas modificações podem causar.

Manutenção Evolutiva

Manutenção Evolutiva é um tipo de manutenção de software que se concentra em melhorar ou aprimorar a funcionalidade do software existente. Ela é realizada para atender às necessidades dos usuários finais e acompanhar as mudanças do ambiente em que o software opera.

Na Manutenção Evolutiva, são realizadas atividades como análise de requisitos, design, desenvolvimento, teste e implantação de novas funcionalidades ou melhorias no software existente. Essas atividades são executadas com o objetivo de melhorar a qualidade e a eficiência do software, tornando-o mais útil e adequado às necessidades dos usuários.

A Manutenção Evolutiva é essencial para garantir a relevância e a competitividade de um software no mercado. Ela pode incluir a adição de novos recursos, a melhoria de recursos existentes, a otimização do desempenho, a atualização de tecnologias e a correção de bugs.

É importante destacar que a Manutenção Evolutiva deve ser planejada e executada com cuidado, de forma a garantir que o software permaneça confiável, seguro e estável após as mudanças realizadas. Além disso, as mudanças devem ser documentadas adequadamente para facilitar a manutenção futura do software.

Importância

- É uma atividade que pode ser realizada em qualquer fase do ciclo de vida do software, desde que haja uma necessidade ou oportunidade de melhorar a funcionalidade do software existente.
- Pode ser iniciada a partir de uma solicitação dos usuários, de uma mudança nas condições do ambiente de negócios ou de uma atualização tecnológica.
- Antes de iniciar a Manutenção Evolutiva, é importante realizar uma análise cuidadosa dos requisitos do software existente, para garantir que as mudanças propostas sejam adequadas e atendam às necessidades dos usuários.
- Durante a Manutenção Evolutiva, é importante seguir boas práticas de desenvolvimento de software, como o uso de padrões de projeto, testes de unidade e integração, documentação adequada e controle de versão.
- A Manutenção Evolutiva pode envolver a modificação do código-fonte existente, a adição de novos componentes de software ou a substituição de componentes obsoletos.
- Uma vez concluída a Manutenção Evolutiva, é importante testar e validar o software para garantir que as mudanças realizadas funcionem corretamente e não causem problemas inesperados.
- A Manutenção Evolutiva pode ajudar a prolongar a vida útil do software existente, reduzir os custos de desenvolvimento e melhorar a satisfação dos usuários finais.

Manutenção Evolutiva pode ter seus custos associados, pois envolve o desenvolvimento de novas funcionalidades ou melhorias no software existente. Os custos podem variar dependendo do escopo da manutenção evolutiva, da complexidade das mudanças necessárias e do tempo necessário para realizá-las.

Vantagens

- Melhoria da funcionalidade: a Manutenção Evolutiva pode adicionar novas funcionalidades ou melhorar as funcionalidades existentes, tornando o software mais útil e eficiente.
- Maior satisfação do usuário: ao atender às necessidades dos usuários finais, a Manutenção Evolutiva pode melhorar a satisfação dos usuários com o software.
- Prolongamento da vida útil do software: a Manutenção Evolutiva pode estender a vida útil do software existente, evitando a necessidade de desenvolver um novo software do zero.
- Redução de custos: a Manutenção Evolutiva pode ser uma opção mais econômica do que o desenvolvimento de um novo software, pois aproveita o investimento já feito no software existente.
- Acompanhamento das mudanças do ambiente: a Manutenção Evolutiva pode garantir que o software esteja atualizado em relação às mudanças do ambiente em que opera, como novas tecnologias, leis ou regulamentos

Desafios

- Risco de introduzir erros: as mudanças feitas durante a Manutenção Evolutiva podem introduzir novos erros ou causar problemas inesperados no software existente.
- Aumento da complexidade: a Manutenção Evolutiva pode aumentar a complexidade do software existente, tornando-o mais difícil de entender e manter.
- Custo: como mencionado anteriormente, a Manutenção Evolutiva pode ter custos associados, especialmente se a complexidade das mudanças necessárias for alta.

Em resumo, a Manutenção Evolutiva pode ser benéfica para melhorar a qualidade e a eficiência do software existente, mas é importante avaliar cuidadosamente os custos e benefícios antes de iniciar qualquer trabalho de Manutenção Evolutiva.

Ciclo PDCA

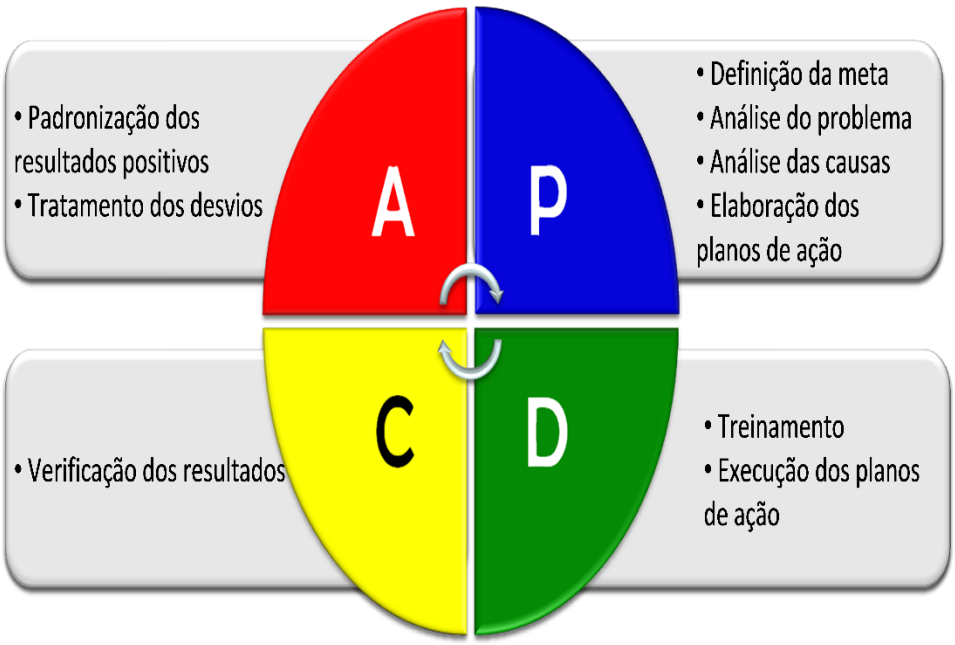
O Ciclo PDCA (ou Ciclo de Deming, em homenagem ao seu criador, o estatístico W. Edwards Deming) é um modelo de gestão que consiste em quatro etapas: Planejar (Plan), Executar (Do), Verificar (Check) e agir (Act).

O objetivo do Ciclo PDCA é melhorar continuamente os processos e produtos de uma organização, por meio da análise sistemática de seus resultados e da adoção de ações corretivas para solucionar eventuais problemas identificados.

Etapas

1. Planejar (Plan): nesta etapa, é definido o objetivo a ser alcançado, os processos necessários para alcançá-lo e as medidas de desempenho que serão utilizadas para avaliar os resultados. É importante que os objetivos e processos sejam claros e bem definidos, para garantir que as ações posteriores sejam efetivas.
2. Executar (Do): nesta etapa, os processos planejados são executados. É importante que as atividades sejam executadas de acordo com o planejado, para garantir que os resultados possam ser avaliados com precisão na etapa seguinte.
3. Verificar (Check): nesta etapa, são coletados e analisados os dados e informações relevantes para avaliar os resultados alcançados. É importante que os dados sejam precisos e relevantes, para que a avaliação seja correta e as ações corretivas possam ser tomadas adequadamente.
4. Agir (Act): nesta etapa, as ações corretivas são identificadas e implementadas, com o objetivo de melhorar continuamente os processos e produtos da organização. É importante que as ações corretivas sejam efetivas e monitoradas para garantir que os problemas identificados sejam resolvidos.

O Ciclo PDCA é uma ferramenta simples, mas poderosa, que pode ser aplicada em uma variedade de contextos, desde a gestão de processos de produção até a melhoria da qualidade de produtos ou serviços. Ao aplicar o Ciclo PDCA, as organizações podem melhorar continuamente seus processos e produtos, aumentar sua eficiência e produtividade e, consequentemente, melhorar sua competitividade no mercado.



Benefícios

- Melhoria contínua dos processos e produtos: o Ciclo PDCA permite identificar problemas e oportunidades de melhoria de forma sistemática, o que possibilita a adoção de ações corretivas e preventivas para melhorar continuamente os processos e produtos da organização.
- Maior eficiência e produtividade: ao identificar e eliminar problemas nos processos, o Ciclo PDCA pode aumentar a eficiência e produtividade da organização, reduzindo custos e tempo de produção.
- Maior satisfação do cliente: ao melhorar continuamente a qualidade dos produtos e serviços oferecidos, o Ciclo PDCA pode aumentar a satisfação do cliente, o que pode levar a um aumento na fidelização e na recomendação da marca.
- Maior competitividade: ao melhorar seus processos e produtos, a organização pode se tornar mais competitiva no mercado, atraindo mais clientes e aumentando sua participação no mercado.

Custos

- Investimento em treinamento: para que o Ciclo PDCA seja implementado de forma efetiva, é necessário que os colaboradores da organização estejam capacitados para identificar problemas e oportunidades de melhoria, além de saber como aplicar as técnicas e ferramentas do Ciclo PDCA.
- Investimento em tecnologia: em alguns casos, pode ser necessário investir em tecnologia para coletar e analisar os dados necessários para a implementação do Ciclo PDCA.

Além disso, um possível malefício do Ciclo PDCA é o risco de se tornar um processo burocrático e repetitivo, sem que haja uma real mudança nos processos e produtos da organização. Para evitar esse risco, é importante que a organização esteja comprometida com a melhoria contínua e que o Ciclo PDCA seja aplicado de forma sistemática e efetiva.

Conclusão

A manutenção de software é uma atividade muito importante na prática das empresas de desenvolvimento de software. Apesar de sua importância, ainda é uma prática malvista pela maioria dos profissionais, pouco estudada e entendida.