Click to prove you're human

Pular para o conteúdo principal O ciclo de vida de desenvolvimento de software (SDLC) é o processo econômico e rápido que as equipes de desenvolvimento usam para projetar e criar software de alta qualidade. O objetivo do SDLC é minimizar os riscos do projeto por meio do planejamento antecipado, para que o software atenda às expectativas do cliente durante e depois da produção. Essa metodologia descreve uma série de etapas que dividem o processo de desenvolvimento de software em tarefas que você pode atribuir, concluir e avaliar. O gerenciamento do desenvolvimento de software pode ser desafiador, devido às alterações dos requisitos, as atualizações na tecnologia e a colaboração multifuncional. A metodologia do ciclo de vida de desenvolvimento de software (SDLC) oferece uma estrutura de gerenciamento sistemática, com produtos específicos em cada estágio do processo de desenvolvimento do software. Como resultado, todos os participantes do processo concordam antecipadamente com as metas e os requisitos do desenvolvimento do software e têm um plano para alcançar essas metas. Alguns benefícios do SDLC: Maior visibilidade do processo de desenvolvimento para todas as partes envolvidas Estimativa, planejamento e programação eficientes Melhor gerenciamento de riscos e estimativa de custos Entrega sistemática do software e maior satisfação do cliente O ciclo de vida do desenvolvimento de software (SDLC) destaca várias tarefas necessárias para criar uma aplicação de software. O processo de desenvolvimento passa por vários estágios, à medida que os desenvolvedores adicionam novos recursos e corrigem bugs no software. Os detalhes dos processos do SDLC variam para diferentes equipes. Porém, destacamos abaixo algumas fases comuns do SDLC: Planejamento A fase de planejamento normalmente inclui tarefas como a análise do custo-benefício, a programação, a estimativa e a alocação de recursos. A equipe de desenvolvimento coleta requisitos de várias partes envolvidas, como clientes, especialistas internos e externos e gerentes, a fim de criar um documento de especificação de requisitos do software. O documento define as expectativas e as metas comuns que ajudarão no planejamento do projeto. A equipe estima os custos, cria uma programação e tem um plano detalhado para atingir suas metas. Projeto Na fase de projeto, os engenheiros de software analisam os requisitos e identificam as melhores soluções para criar o software. Por exemplo, eles poderão considerar a integração de módulos pré-existentes, fazer escolhas tecnológicas e identificar ferramentas de desenvolvimento. Eles analisarão qual a melhor forma de integrar o novo software às infraestruturas de TI existentes que a organização já tiver. Implementação Na fase de implementação, a equipe de desenvolvimento codifica o produto. Eles analisam os requisitos para identificar as tarefas de codificação menores que podem realizar diariamente para alcançar o resultado final. Teste A equipe de desenvolvimento combina testes manuais e automatizados para identificar bugs no software. A análise de qualidade inclui testar o software para identificar erros e verificar se ele atende aos requisitos do cliente. Como muitas equipes testam o código que desenvolvem, a fase de teste pode ocorrer em paralelo à fase de desenvolvimento. Implantação Quando as equipes desenvolvem software, eles criam e testam o código em uma cópia do software diferente daquela à qual os usuários têm acesso. O software usado pelo cliente é considerado em produção, enquanto outras cópias são consideradas como o ambiente de compilação ou ambiente de teste. O uso de ambientes de compilação e produção separados garante que os clientes possam continuar a usar o software, mesmo quando ele estiver sendo alterado ou atualizado. A fase de implantação inclui várias tarefas para mover a cópia mais recente para o ambiente de produção, como empacotamento, configuração de ambientes e instalação. Manutenção Na fase de manutenção, entre outras tarefas, a equipe corrige bugs, soluciona problemas do cliente e gerencia as alterações do software. Além disso, a equipe monitora a performance geral do sistema, a segurança e a experiência do usuário para identificar novas formas de melhorar o software existente. Um modelo de ciclo de vida de desenvolvimento de software (SDLC) apresenta conceitualmente o SDLC de forma organizada para ajudar as empresas a implementá-lo. Diferentes modelos organizam as fases do SDLC em ordens cronológicas variadas, a fim de otimizar o ciclo de desenvolvimento. Analisaremos abaixo alguns modelos de SDLC mais usados: Cascata O modelo cascata organiza todas as fases sequencialmente, de forma que cada fase nova dependa do resultado da fase anterior. Conceitualmente, o design flui de uma fase para a outra, como uma cascata. Prós e contras O modelo cascata proporciona disciplina para o gerenciamento de projetos e oferece um resultado tangível ao final de cada fase. Porém, não é possível fazer alterações depois que uma fase for concluída, pois as alterações podem afetar o tempo de entrega, os custos e a qualidade do software. Portanto, esse modelo é mais adequado para pequenos projetos de desenvolvimento de software, nos quais é mais fácil organizar e gerenciar tarefas e os requisitos podem ser predefinidos com precisão. Iterativo O processo iterativo sugere que as equipes comecem o desenvolvimento de software com um pequeno subconjunto de requisitos. Com o passar do tempo, eles aprimoram as versões iterativamente até que o software completo esteja pronto para produção. A equipe produz uma nova versão do software ao final de cada iteração. Prós e contras A identificação e o gerenciamento de riscos são fáceis, uma vez que os requisitos podem ser alterados entre as iterações. Porém, a repetição de ciclos pode levar à alteração do escopo e a subestimação de recursos. Espiral O modelo espiral combina os pequenos ciclos repetidos do modelo iterativo com o fluxo sequencial linear do modelo cascata, a fim de priorizar a análise de riscos. O modelo espiral pode ser usado para garantir o lançamento e o aprimoramento gradual do software, com a criação de protótipos em cada fase. Prós e contras O modelo espiral é adequado para projetos grandes e complexos, que exigem alterações frequentes. Porém, ele pode ter um custo alto para projetos menores com escopo limitado. Ágil O modelo ágil organiza as fases do SDLC em vários ciclos de desenvolvimento. A equipe itera as fases rapidamente, fazendo apenas alterações pequenas e incrementais ao software em cada ciclo. Eles avaliam continuamente os requisitos, planos e resultados para que possam responder rapidamente às alterações. O modelo ágil é iterativo e incremental, o que o torna mais eficiente do que outros modelos de processo. Prós e contras Os ciclos de desenvolvimento rápidos ajudam as equipes a identificar e resolver questões antecipadamente em projetos complexos, antes que elas se tornem problemas significativos. Eles também podem envolver clientes e outras partes interessadas para obter feedback em todo o ciclo de vida do projeto. Porém, a dependência exagerada do feedback de clientes pode levar a alterações excessivas no escopo ou ao encerramento prematuro do projeto. Nos métodos tradicionais de desenvolvimento de software, o teste de segurança era um processo separado do ciclo de vida do desenvolvimento de software (SDLC). A equipe de segurança descobria as falhas de segurança somente depois de criar o software. Isso levava a um alto número de bugs que permaneciam ocultos e aumentavam os riscos de segurança. Hoje, a maioria das equipes reconhece que a segurança é uma parte integral do ciclo de vida do desenvolvimento de software. Você pode administrar a segurança no SDLC adotando as práticas de DevSecOps e conduzindo avaliações de segurança durante todo o processo do SDLC. DevSecOps DevSecOps é a prática de integrar testes de segurança em todas as etapas do processo de desenvolvimento de software. Ele inclui ferramentas e processos que incentivam a colaboração entre desenvolvedores, especialistas em segurança e equipes operacionais para criar software resiliente às ameaças modernas. Além disso, o DevSecOps garante que as atividades de segurança, como revisão de código, análise da arquitetura e testes de penetração, sejam parte integrante das iniciativas de desenvolvimento. Leia sobre os Parceiros com competência em DevOps da AWS» O termo ciclo de vida de desenvolvimento de software (SDLC) é bastante usado em tecnologia para se referir a todo o processo de inovação e suporte tecnológico. Veja abaixo outros termos semelhantes. Ciclo de vida do desenvolvimento de sistemas A abreviação SDLC pode às vezes se referir ao ciclo de vida dos sistemas de desenvolvimento, o processo para o planejamento e a criação de um sistema de TI. O sistema consiste normalmente de vários componentes de software e hardware que trabalham juntos para executar funções complexas. Ciclo de vida de desenvolvimento de software comparado ao ciclo de vida de desenvolvimento de sistemas O ciclo de vida do desenvolvimento de software lida somente com o desenvolvimento e o teste de componentes do software. Por outro lado, o desenvolvimento de sistemas é um superconjunto mais amplo que inclui a configuração e o gerenciamento de software, hardware, pessoas e processos que formam um sistema. Isso pode incluir tarefas como treinamento organizacional e políticas de gerenciamento de alterações que não fazem parte da sombrinha de desenvolvimento de software. Gerenciamento do ciclo de vida da aplicação O gerenciamento do ciclo de vida da aplicação (ALM) é a criação e manutenção de uma aplicação de software até que ela não seja mais necessária. Ele envolve vários processos, ferramentas e pessoas trabalhando juntas para gerenciar todos os aspectos do ciclo de vida, como ideias, projeto e desenvolvimento, testes, produção, suporte e eventual redundância. SDLC comparado ao ALM O SDLC descreve a fase de desenvolvimento da aplicação com mais detalhes. Ele faz parte do ALM. O ALM inclui todo o ciclo de vida da aplicação e continua além do SDLC. O ALM pode ter vários SDLCs durante o ciclo de vida de uma aplicação. As ferramentas da AWS para desenvolvedores contam com vários serviços que inserem eficiência ao ciclo de vida do desenvolvimento de software (SDLC). Veja alguns exemplos: O Amazon CodeGuru é uma ferramenta para desenvolvedores que fornece recomendações inteligentes para melhorar a qualidade do código e identificar as linhas de código mais caras de uma aplicação. Integre CodeGuru ao seu fluxo de trabalho de desenvolvimento de software existente para automatizar as revisões de código e monitorar continuamente a performance das aplicações no ambiente de produção. O AWS CodePipeline é um serviço totalmente gerenciado que ajuda a automatizar os ciclos de lançamento para oferecer atualizações rápidas e confiáveis de aplicações e infraestruturas. O AWS CodeBuild é um serviço totalmente gerenciado que compila código-fonte, executa testes e produz pacotes de software prontos para implantação. O CodeBuild escala continuamente e processa múltiplas compilações ao mesmo tempo, o que evita que elas fiquem esperando em uma fila. O Amazon Elastic Container Service (Amazon ECS) é um serviço totalmente gerenciado que facilita a implantação, o gerenciamento e a escala de aplicações em contêineres. Além disso, o Amazon Managed Grafana é um serviço totalmente gerenciado para Grafana de código aberto desenvolvido em colaboração com a Grafana Labs. Grafana é uma famosa plataforma de análise de código aberto que permite que você consulte, visualize, gere alertas e entenda suas métricas independentemente de onde elas estiverem armazenadas. Com um upgrade opcional para o Grafana Enterprise, você pode acessar mais plug-ins de terceiros que fornecem recursos de monitoramento SDLC, como ServiceNow e Atlassian Jira. Usando esses plug-ins, você pode levar os detalhes dos incidentes e os produtos do SDLC para o Amazon Managed Grafana. Assim, você pode rastrear o status dos incidentes, obter as solicitações e confirmações de códigos e monitorar as versões de software, juntamente com os dados de performance e integridade da aplicação, em um só local. Comece a usar o SDLC na AWS criando uma conta gratuita hoje mesmo!