 MELHOR SERVIÇO DE HOSPEDAGEM E E-MAIL  
  
Gerenciamento de código fonte (do Inglês Source Code Manegement – SCM) é usado para rastrear modificações em um repositório de código fonte de um projeto de desenvolvimento de software.  
O SCM monitora o histórico de mudanças em execução em uma base de código e ajuda a resolver conflitos ao mesclar atualizações de vários programadores. SCM também é sinônimo de controle de versão.  
À medida que os projetos de software crescem em linhas de código e número de colaboradores, os custos de sobrecarga de comunicação e complexidade de gerenciamento também aumentam.  
SCM é uma ferramenta crítica para melhorar organizacionalmente as complexidades crescentes de desenvolvimento de softwares.  
Vídeo aula do curso Git Básico e Essencial  
  
Índice  ocultar   
  
1. A importância das ferramentas de gerenciamento de código fonte  
  
1.1. SCM  
  
2. Os benefícios do gerenciamento de código fonte  
  
3. Resumo  
  
3.1. Artigos relacionados  
  
A importância das ferramentas de gerenciamento de código fonte  
  
Quando vários desenvolvedores estão trabalhando em uma base de código de maneira colaborativa, é comum fazer alterações em uma parte compartilhada do código.  
Desenvolvedores separados podem estar trabalhando em um recurso aparentemente isolado, no entanto, esse recurso pode usar um módulo de código compartilhado.  
Portanto, o desenvolvedor 1 que está trabalhando no recurso 1 pode fazer algumas edições e descobrir depois que o desenvolvedor 2 que está trabalhando no recurso 2 tem edições conflitantes no mesmo arquivo ou conjunto de arquivos.  
Sem a adoção do SCM, este é um cenário de pesadelo.  
Por exemplo: os desenvolvedores podem editar arquivos de texto diretamente e movê-los para locais remotos usando FTP ou outros protocolos.  
Imagine que o Desenvolvedor 1 faz edições, na sequência, o Desenvolvedor 2, sem saber, salva suas edições sobre o trabalho do Desenvolvedor 1 e apaga as alterações.  
O papel do SCM como mecanismo de proteção contra este cenário específico é conhecido como Controle de Versão.  
Leia o artigo Controle de Versão – O que é? para aprofundar seu conhecimento neste conceito.  
  
SCM  
  
O Gerenciamento de código fonte trouxe proteções de controle de versão para evitar a perda de trabalho devido à substituição de conflitos.  
Essas proteções funcionam rastreando as alterações de cada desenvolvedor individualmente, identificando áreas de conflito e evitando gravações sobrepostas de porções de código.  
O SCM então comunicará esses pontos de conflito aos desenvolvedores para que eles possam revisar e resolver com segurança antes de mesclar suas alterações na versão definitiva.  
Este mecanismo básico de prevenção de conflitos  tem o efeito colateral, benéfico, de fornecer comunicação passiva para a equipe de desenvolvimento.  
Desta forma a equipe pode acompanhar e discutir o trabalho em andamento que o gerenciamento de código-fonte está monitorando.  
O SCM rastreia todo o histórico de mudanças na base de código. Isto permite que os desenvolvedores examinem e revisem as edições que podem ter introduzido bugs (erros) ou regressões de funcionalidades. Isto é, perda de trabalho quando uma alteração é sobreposta com uma versão antiga de código.  
  
￼Exemplo de um histórico de projeto gerenciado por Git  
  
Os benefícios do gerenciamento de código fonte  
  
Além do controle de versão, o SCM fornece um conjunto de outros recursos úteis para tornar o desenvolvimento de código colaborativo uma experiência mais amigável ao usuário.  
Depois que o SCM começa a rastrear todas as alterações em um projeto ao longo do tempo, ele cria um registro histórico detalhado da vida do projeto.  
Este registro histórico pode então ser usado para “desfazer” as alterações na base de código. O SCM pode reverter instantaneamente a base de código para um ponto anterior no tempo.  
Isso é extremamente valioso para evitar regressões nas atualizações e desfazer erros.  
O arquivo de Versionamento de cada mudança durante o tempo de vida de um projeto, fornece um registro valioso para as notas (anotações, comentários) de versão de lançamento de um projeto.  
Um registro de histórico de SCM limpo e mantido com boa descrição das alterações pode ser usado alternadamente como notas de versão.  
Isso oferece uma visão e transparência sobre o andamento de um projeto que pode ser compartilhado com usuários finais ou equipes que não são de desenvolvimento de software.  
A gestão de código fonte reduzirá a sobrecarga de comunicação de uma equipe e aumentará a velocidade de liberação de novas versões.  
Sem o SCM, o desenvolvimento é mais lento porque os contribuidores precisam fazer esforço extra para planejar uma sequência não sobreposta de desenvolvimento para lançamento.  
Com o versionamento do software, os desenvolvedores podem trabalhar independentemente em ramos separados no desenvolvimento de recursos, eventualmente mesclando-os.  
Ramos são linhas de tempo isoladas, chamadas branch.  
Em geral, o gerenciamento de código fonte é uma grande ajuda para as equipes de engenharia de software, que reduz os custos de desenvolvimento, permitindo que os recursos de engenharia sejam executados com mais eficiência.  
SCM é essencial na era moderna do desenvolvimento de software.  
  
Equipes profissionais usam controle de versão e sua equipe também deve utilizar.  
  
Resumo  
  
Versionamento pelo gerenciamento de código fonte é uma ferramenta inestimável para o desenvolvimento de software moderno.  
  
As melhores equipes de software praticam versionamento de software e sua equipe também deve adotar.  
  
O gerenciamento de código fonte é muito fácil de configurar em um novo projeto e o retorno do investimento é alto.  
  
Para aprender mais sobre versionamento de software, gerenciamento de código fonte, e as ferramentas usadas para executá-lo, como Git, Github e Gitlab, leia os meus artigos:  
  
Cache temporário das credenciais GIT – username e password  
  
Existem situações em que você precisa trabalhar com um repositório GIT usando o protocolo HTTPs e não pode armazenar permanentemente as credenciais de usuário na configuração local. Uma solução é armazenar em um cache temporário as credenciais (username e password – senha) por um tempo limitado. git-credential-cache é o auxiliar que vamos usar para armazenar… Continue a ler »

nviado do meu Galaxy O MELHOR SERVIÇO DE HOSPEDAGEM E E-MAILGerenciamento de código fonte (do Inglês Source Code Manegement – SCM) é usado para rastrear modificações em um repositório de código fonte de um projeto de desenvolvimento de software.O SCM monitora o histórico de mudanças em execução em uma base de código e ajuda a resolver conflitos ao mesclar atualizações de vários programadores. SCM também é sinônimo de controle de versão.À medida que os projetos de software crescem em linhas de código e número de colaboradores, os custos de sobrecarga de comunicação e complexidade de gerenciamento também aumentam.SCM é uma ferramenta crítica para melhorar organizacionalmente as complexidades crescentes de desenvolvimento de softwares.Vídeo aula do curso Git Básico e EssencialÍndice  ocultar 1. A importância das ferramentas de gerenciamento de código fonte1.1. SCM2. Os benefícios do gerenciamento de código fonte3. Resumo3.1. Artigos relacionadosA importância das ferramentas de gerenciamento de código fonteQuando vários desenvolvedores estão trabalhando em uma base de código de maneira colaborativa, é comum fazer alterações em uma parte compartilhada do código.Desenvolvedores separados podem estar trabalhando em um recurso aparentemente isolado, no entanto, esse recurso pode usar um módulo de código compartilhado.Portanto, o desenvolvedor 1 que está trabalhando no recurso 1 pode fazer algumas edições e descobrir depois que o desenvolvedor 2 que está trabalhando no recurso 2 tem edições conflitantes no mesmo arquivo ou conjunto de arquivos.Sem a adoção do SCM, este é um cenário de pesadelo.Por exemplo: os desenvolvedores podem editar arquivos de texto diretamente e movê-los para locais remotos usando FTP ou outros protocolos.Imagine que o Desenvolvedor 1 faz edições, na sequência, o Desenvolvedor 2, sem saber, salva suas edições sobre o trabalho do Desenvolvedor 1 e apaga as alterações.O papel do SCM como mecanismo de proteção contra este cenário específico é conhecido como Controle de Versão.Leia o artigo Controle de Versão – O que é? para aprofundar seu conhecimento neste conceito.SCMO Gerenciamento de código fonte trouxe proteções de controle de versão para evitar a perda de trabalho devido à substituição de conflitos.Essas proteções funcionam rastreando as alterações de cada desenvolvedor individualmente, identificando áreas de conflito e evitando gravações sobrepostas de porções de código.O SCM então comunicará esses pontos de conflito aos desenvolvedores para que eles possam revisar e resolver com segurança antes de mesclar suas alterações na versão definitiva.Este mecanismo básico de prevenção de conflitos  tem o efeito colateral, benéfico, de fornecer comunicação passiva para a equipe de desenvolvimento.Desta forma a equipe pode acompanhar e discutir o trabalho em andamento que o gerenciamento de código-fonte está monitorando.O SCM rastreia todo o histórico de mudanças na base de código. Isto permite que os desenvolvedores examinem e revisem as edições que podem ter introduzido bugs (erros) ou regressões de funcionalidades. Isto é, perda de trabalho quando uma alteração é sobreposta com uma versão antiga de código.￼Exemplo de um histórico de projeto gerenciado por GitOs benefícios do gerenciamento de código fonteAlém do controle de versão, o SCM fornece um conjunto de outros recursos úteis para tornar o desenvolvimento de código colaborativo uma experiência mais amigável ao usuário.Depois que o SCM começa a rastrear todas as alterações em um projeto ao longo do tempo, ele cria um registro histórico detalhado da vida do projeto.Este registro histórico pode então ser usado para “desfazer” as alterações na base de código. O SCM pode reverter instantaneamente a base de código para um ponto anterior no tempo.Isso é extremamente valioso para evitar regressões nas atualizações e desfazer erros.O arquivo de Versionamento de cada mudança durante o tempo de vida de um projeto, fornece um registro valioso para as notas (anotações, comentários) de versão de lançamento de um projeto.Um registro de histórico de SCM limpo e mantido com boa descrição das alterações pode ser usado alternadamente como notas de versão.Isso oferece uma visão e transparência sobre o andamento de um projeto que pode ser compartilhado com usuários finais ou equipes que não são de desenvolvimento de software.A gestão de código fonte reduzirá a sobrecarga de comunicação de uma equipe e aumentará a velocidade de liberação de novas versões.Sem o SCM, o desenvolvimento é mais lento porque os contribuidores precisam fazer esforço extra para planejar uma sequência não sobreposta de desenvolvimento para lançamento.Com o versionamento do software, os desenvolvedores podem trabalhar independentemente em ramos separados no desenvolvimento de recursos, eventualmente mesclando-os.Ramos são linhas de tempo isoladas, chamadas branch.Em geral, o gerenciamento de código fonte é uma grande ajuda para as equipes de engenharia de software, que reduz os custos de desenvolvimento, permitindo que os recursos de engenharia sejam executados com mais eficiência.SCM é essencial na era moderna do desenvolvimento de software.Equipes profissionais usam controle de versão e sua equipe também deve utilizar.ResumoVersionamento pelo gerenciamento de código fonte é uma ferramenta inestimável para o desenvolvimento de software moderno.As melhores equipes de software praticam versionamento de software e sua equipe também deve adotar.O gerenciamento de código fonte é muito fácil de configurar em um novo projeto e o retorno do investimento é alto.Para aprender mais sobre versionamento de software, gerenciamento de código fonte, e as ferramentas usadas para executá-lo, como Git, Github e Gitlab, leia os meus artigos:Cache temporário das credenciais GIT – username e passwordExistem situações em que você precisa trabalhar com um repositório GIT usando o protocolo HTTPs e não pode armazenar permanentemente as credenciais de usuário na configuração local. Uma solução é armazenar em um cache temporário as credenciais (username e password – senha) por um tempo limitado. git-credential-cache é o auxiliar que vamos usar para armazenar… Continue a ler »