Comparando os 2 controladores de versão o Git não é melhor que o Subversion. Mas também não é pior. É diferente. A principal diferença é que ela é descentralizada.

Com o Subversion, você tem um Problema: O Repositório SVN pode estar em um local que você não pode alcançar (na sua empresa, e você não tem internet no momento), você não pode confirmar. Se você quiser fazer uma cópia do seu código, você tem que literalmente copiá-lo / colá-lo.

Com o Git, você não tem esse problema. Sua cópia local é um repositório, e você pode se comprometer com ela e obter todos os benefícios do controle de origem. Quando você recuperar a conectividade com o repositório principal, poderá confirmar isso.

O Git REALMENTE brilha quando você é descentralizado. Eu tenho um servidor em casa e um laptop na estrada, e o SVN simplesmente não funciona bem aqui. Com o SVN, não posso ter controle de origem local se não estiver conectado ao repositório (sim, eu sei sobre o SVK ou sobre como copiar o repositório). Com o Git, esse é o modo padrão de qualquer maneira. É um comando extra embora (git commit commits localmente, enquanto o git push origin master envia o branch master para o remote chamado "origin").

Como dito acima: o Git adiciona complexidade. Dois modos de criar repositórios, check-out vs. clone, commit vs. push ... Você precisa saber quais comandos funcionam localmente e quais funcionam com "o servidor" (estou assumindo que a maioria das pessoas ainda gosta de um "repositório mestre" central) ).

Além disso, o ferramental ainda é insuficiente, pelo menos no Windows. Sim, há um AddIn do Visual Studio, mas ainda uso git bash com msysgit.

O SVN tem a vantagem de ser MUITO mais simples de aprender: existe o seu repositório, todas as mudanças em relação a ele, se você sabe como criar, confirmar e finalizar e está pronto para pegar coisas como ramificação, atualização etc. em.

O Git tem a vantagem de ser MUITO mais adequado se alguns desenvolvedores nem sempre estiverem conectados ao repositório principal. Além disso, é muito mais rápido que o SVN. E pelo que ouvi, ramificar e mesclar o suporte é muito melhor (o que é de se esperar, já que essas são as principais razões pelas quais ele foi escrito).

Isso também explica por que ele ganha tanto burburinho na Internet, já que o Git é perfeitamente adequado para projetos de código aberto: apenas bifurque-o, confirme suas alterações em seu próprio fork e peça ao mantenedor original do projeto para fazer as alterações. Com o Git, isso simplesmente funciona. Realmente, experimente no Github, é mágico.  
  
  
GIT VS SVN  
SVN:  
Sistema de controle de versão centralizado:  
repositório autoritativo central

GIT:

Todos tem seu próprio repositório

você obtém a rede de segurança de check-ins frequentes sem ter que se preocupar com "quebrar a compilação"

os clientes podem fazer alterações nos repositórios e essas alterações serão locais para eles, a menos que sejam sincronizados com outra pessoa

dependendo dos requisitos, o Git também oferece um repositório centralizado

**controle de fonte online:**

SVN:

o usuário deve estar online para se comprometer com o repositório a partir da cópia de trabalho

**Controle de fonte off-line:**

GIT:

os clientes podem confirmar alterações em seus repositórios localizados como novas revisões mesmo estando offline

**Velocidade:**

SVN:  
Maior tempo para armazenar, porque todos os dados são armazenados em um repositório centralizado

GIT:

Extremamente rápida, a cópia completa dos dados fica armazenada localmente no sistema do cliente

muito menos tempo de resposta da rede

**SVN:**

A “ajuda” do SVN é mais organizada

**GIT:**

há algum tempo perdido, pois é difícil obter uma referência rápida da pesquisa do git

**MAIS ESPAÇO DE ARMAZENAMENTO**

SVN:

Duas cópias de um arquivo no diretório de trabalho do SVN

Uma cópia é usada para armazenar o trabalho real, enquanto a outra cópia oculta em .svn / contém as informações usadas para auxiliar as operações (status e confirmação)

Quando há muitos documentos, há um enorme impacto do espaço em disco no SVN em comparação com o GIT

**Menos espaço de armazenamento**

GIT:

Tem uma memória eficiente porque o formato de arquivo dos dados está compactado

O Git tem um pequeno arquivo de índice para armazenar as informações relacionadas a um arquivo em particular

SVN:  
Criar e trabalhar em um branch / merge é complexo e difícil de usar

GIT:

Simples e fácil de usar branch / merge

O diretório de trabalho de um desenvolvedor é em si um branch

SVN:

**Commits sequenciais**

Os dados são perdidos quando confirmações simultâneas de duas ou mais cópias de trabalho são realizadas

GIT:

**Sem commits sequenciais**

Grande número de usuários pode colocar dados no mesmo repositório

Não precisa se preocupar com perda de dados ou fusão imediata de outras alterações