Internal information
Init

Prepared by	Document
Mikko Koivunalho <mikko.koivunalho@iki.fi>	Jämförelse: Subversion och Git
Approved by	Date
	16 juni 2014

Jämförelse: Subversion och Git

Mikko Koivunalho <mikko.koivunalho@iki.fi>

16 juni 2014

Innehåll

Internal information
Init

Prepared by	Document
Mikko Koivunalho <mikko.koivunalho@iki.fi>	Jämförelse: Subversion och Git
Approved by	Date
	16 juni 2014

1 Motivering

Frågan om övergång från *Subversion* till *Git* (eller decentraliserad versionshantering i allmänhet) har blivit allt viktigare. Det finns många orsaker till det (exempelvis används *Git* av stora öppen-källkodprojekt, speciellt *Linux*) men den antagligen viktigaste är att många unga utvecklare, som har fått sina första programmeringserfarenheter från öppen-källkodprojekt (av vilka många använder *Git*, speciellt *GitHub*), frågar efter *Git*.

Vi vill göra en undersökning av vilka orsaker som rättfärdigar övergången från *Subversion* till *Git* (speciellt i mitten av projekt) och vilka inte.

Undertecknad har använt Subversion, Git och *Bazaar* dagligen och har skapat Subversion-*repositories* två gånger.

2 Decentraliserad versionshantering

Centraliserad versionshantering är det klassiska sättet att arrangera ett *repository*, d.v.s. en central plats för all kod att ligga i. När man börjar utveckla koden, gör man först en lokal kopia av all kod som behövs, alltså *checkout* av projektets kod eller bara den del man behöver. Användningsmodellen är alltså *client-server*.

I decentraliserad (d.v.s. distribuerad) versionshantering finns inget centralt *repository*. Användningsmodellen är *peer-to-peer*. Varje användare har hela kodbasen och dess historia som sitt lokala *repository*. Användarna uppdaterar sina kodbaser från andra användares kod. Synkronisering görs som *patching* där man väljer vilka patchar man vill ta in.

2.1 Fördelar med decentraliserad versionshantering

Några ofta uttalade fördelar(3) med decentraliserad versionshantering jämfört med centraliserad är:

1. Man kan jobba även utan nätverk.
2. De flesta operationer är snabbare p.g.a. att inget nätverk behövs.
3. Det finns ingen *single point of failure*, alltså är kodbasen alltid trygg eftersom alla användare har hela kodbasen lokalt.
4. Det är lättare att göra en *fork* av projektet.

5. Det är lätt för vem som helst att delta i projektet eftersom man inte behöver ha tillåtelse från projektets ledare innan man kan skicka in sin kod.

Joel Spolsky, grundare av Fog Creek Software, som specialiserar sig på *project management*-mjukvara (inkl. Kiln, ett versionshanteringssystem), har sagt i sin blog att distribuerad versionshantering är *"possibly the biggest advance in software development technology in the ten years I've been writing articles here. Or, to put it another way, I'd go back to C++ before I gave up on Mercurial"*.(2)

3 Introduktion

3.1 Subversion

Subversion är en efterträdare till *CVS* (Concurrent Version System) och har ärvt CVS:s position som marknadsledare. Subversion är helt bakåtkompatibel med CVS så alla CVS:s skript är användbara utan modifiering. Subversion är öppen källkod och utvecklas av Apache Software Foundation.

3.2 Git

Git designades och utvecklades 2005 av Linus Torvalds[1] för Linux Kernel-utveckling. Viktigast för Torvalds var snabbhet och säkerhet.(4) Gits design fick inspiration från tidigare *BitKeeper* och *Monotone*. Git är öppen källkod och utvecklas av Junio Hamano.(5)

3.3 Popularitet

Year	Git	Subversion
2009	2.4%	57.5%
2010	6.8%	58.3%
2011	12.8%	51.3%
2012	27.6%	46.0%
2013	36.3%	37.8%

Tabell 1: Results of the Eclipse Community Survey regarding SVN and Git usage.(8, slide 14)

[1] *Linux*:s grundare

Prepared by	Document
Mikko Koivunalho <mikko.koivunalho@iki.fi>	Jämförelse: Subversion och Git
Approved by	Date
	16 juni 2014

Enligt Eclipses Survey har Git blivit allt populärare. Täckande statistik från stora företag finns inte men vi kan använda annan statistik, t.ex. ItJobsWatch som håller koll på vilka teknologier som är efterfrågade.

Year	Permanent positions:		Rank:	
	Git	Subversion	Git	Subversion
2012	1167	3354	263	91
2013	2049	2836	157	107
2014	3605	3265	90	99

Tabell 2: ITJobsWatch: Git & Subversion.(9)

Samma trend visas här: Git har gått förbi Subversion, eller håller på att göra så. Den senaste statistiken från ItJobsWatch visar att under detta år har Subversion efterfrågats något mer, men skillnaden är så liten att den troligen ligger inom felmarginalen.

3.4 Användare

Den allmänna uppfattningen är att medan Subversion håller sig populär i affärsvärlden, är Git mera populär i öppen-källkodsvärlden. Men ITJobsWatch statistik leder oss att tro att Git har börjat bli populär också i affärsvärlden. Tyvärr, berättar samma statistik inte hurdana bolag som börjat använda Git: stora, små eller startupar.

I varje fall måste vi anta att Git följer samma väg som många andra innovationer i IT-industrin: startup-firmor visar vägen, stora bolag kommer senare efter att ny personal har rekryterats från startupar eller öppen-källkodsvärlden.

4 Jämförelse

4.1 Kostnader

Både Git och Subversion är öppen källkod och helt gratis så det finns inga direkta kostnader. Men kostnader kommer från administration, oflexibla arbetssätt, förslösad tid p.g.a. problemen med att använda versionshantering eller lära sig det, eller om programmet är opålitligt.

Prepared by	Document
Mikko Koivunalho <mikko.koivunalho@iki.fi>	Jämförelse: Subversion och Git
Approved by	Date
	16 juni 2014

	Git	Subversion
Maintainer	Junio Hamano	Apache Software Foundation
Repository model	Distributed	Client-server
Concurrency model	Merge	Merge / lock
License	GNU GPL	Apache
Network protocols	custom (git), custom over ssh, HTTP/HTTPS, rsync, email, bundles	custom (svn), custom over ssh, HTTP and SSL (using WebDAV)

Tabell 3: Comparison: Git & Subversion.

4.2 Ibruktagande

Git är lättare att börja använda i ett projekt; det behöver ju inget centralt *repository*. Vem som helst kan skapa ett nytt arkiv efter vilket andra i teamet kopierar det. Detta kallas *pull*. Eftersom det inte finns en central server brukar man ha en ”dummy user”, en plats dit alla utvecklare pushar sin färdiga kod. Denna plats kan ha en spegel i *build monitor*-systemet (för *nightly builds*) eller i ett *continuous integration*-system, t.ex. *Hudson* eller *Jenkins*.

Subversion måste tvärtom ha ett permanent centralarkiv dit alla committar sin kod. Detta arkiv kan ligga antingen på en lokal server eller en fjärrserver. I Subversion kan man också skapa tomma kataloger. Detta kan man inte göra i Git. Att skapa tomma kataloger kan vara till nytta i början av projektet om man skapar hela projektets katalogstruktur på en gång. Denna brist är lätt att ta sig runt i Git genom att lägga till en fil i katalogen, t.ex. *.gitignore*-filen.

4.3 Användning

För bägge mjukvarorna finns det många klienter för olika plattformar, både officiellt stödda och inte, som funkar med GUI eller med CLI i terminalen. Det finns också pluginer för olika editorer och IDE:er. Det är inte inom ramen för denna rapport att ytterligare utreda dessa.

4.4 Administration

Som tidigare berättat i avsnitt 4.2, behöver Git varken administration eller underhåll. Den andra sidan av myntet är att all denna frihet och decentralisering behöver starka och noggranna regler. Man måste hålla koll på att varje utvecklare har committat och pushat koden till rätt branch. Det finns en viktig social aspekt i samband med Git, vi diskuterar detta i avsnitt 4.5.

Prepared by	Document
Mikko Koivunalho <mikko.koivunalho@iki.fi>	Jämförelse: Subversion och Git
Approved by	Date
	16 juni 2014

Subversion följer i stället samma hierarkiska administrativa tradition som CVS. Det vill inte säga att Subversion som verktyg är oflexibelt och ovänligt att använda; det är det inte. Subversions design (som Gits) följer Unix filosofi: *"Unix was not designed to stop its users from doing stupid things, as that would also stop them from doing clever things."* — Doug Gwyn.(10) Men Subversion behöver mera *direkt* administrativt arbete (som Git inte brukar behöva, eller behöva mindre):

- skapa arkiv.
- hantera arkivets server, backup, o.s.v.
- hantera användare, ge och ta bort användarrättigheter.
- extra arbete, t.ex.
  - när man merge:ar brancher.
  - interface:ar med *bug trackers*, *build*-verktyg, och *continuous integration*-plattformar.

4.5 Sociala aspekter

Frågan om versionshanteringsmetod är inte bara teknisk, fast vi ofta skulle vilja tänka på det så. Versionshantering är en av de viktigaste delarna av projekthantering: man använder ju det varje dag. Det är både ett utvecklingsverktyg och en länk mellan utvecklare och ledare. Det bidrar till utvecklingsteamets kultur.

Ben Collins-Sussman, en av Subversions designers, säger att decentraliserad versionshantering funkar dåligt om alla utvecklingsteamets medlemmar inte är på tillräckligt hög nivå. Utvecklare är också människor, och *"want to work privately, in a cave, then spring 'perfect' code on their community, as if no mistakes had ever been made"*.(7) Om man arbetar med centraliserad versionshantering är det inte så lätt att utveckla ensam. Man måste göra commit ganska ofta (firman kan t.ex. kräva en commit varje dag innan man lämnar kontoret), och om versionshanteringen är kopplad till ett *continuous delivery*-system och unit-testerna blir körda automatiskt, visar sig felen snabbare. Collins-Sussman åberopar Googles kultur och mantra: *"don't run from failure—fail often, fail quickly, and learn"*.

4.6 Slutsats

Att välja mellan Subversion och Git beror mindre på deras tekniska egenskaper än på vad man vill uppnå och inom hurdant arbetsområde.

Prepared by	Document
Mikko Koivunalho <mikko.koivunalho@iki.fi>	Jämförelse: Subversion och Git
Approved by	Date
	16 juni 2014

Git är bra för öppen-källkodprojekt eftersom *administrative overhead* är nästan obefintligt. Git är lätt att börja använda, och snabbare.

Subversion är också mycket populär i öppen-källkodsvärlden. I affärsvärlden tycks Subversion ge mera trovärdighet och strukturellt stöd i projekt, vilket kan vara mycket viktigt t.ex. om personalen i projektet ofta byts.

A Alternativ till Git: Mercurial, GNU Bazaar; kommersiell Perforce

Gits mest kända konkurrenter är *Mercurial* och *Bazaar*. Också dessa är gratis. De liknar Git i stort men det finns också skillnader, t.ex. vill båda göra det lättare för Subversion-användare att byta: de använder många liknande kommandon, många *workflows* är samma. Med Bazaar kan man både använda och inte använda ett centralt *repository* i samma projekt.

Perforce är en kommersiell versionshanteringsmjukvara utvecklad av *Perforce Software, Inc*. Man kan använda Perforce-arkiv med dess eget verktyg *p4*-commandot eller med en *Git*-klient. Det är också möjligt att kopiera hel kodbas och fortsätta använda den som en Git-arkiv.

B GitHub

GitHub (`http://www.github.com`) är ett av de mest populära webbhotellen för mjukvaruutvecklingsprojekt som använder Git. Github erbjuder både abonnemang mot avgift för privat bruk och gratis lagring för mjukvaruprojekt med öppen källkod. Efter att ha skapat ett projektförråd (*repository*), kan man använda det från sin egen dator eller IDE med den normala git-klienten. Genom att använda GitHub eller liknande tjänst är det lätt att prova på Git.

C Byte från centraliserad till decentraliserad versionshantering

Som berättat ovan, är frågan inte bara teknisk. Vi är slavar under våra vanor. Bytet kan betyda en stor kulturförändring. Om idén kommer från företagsledningen eller utanför teamet, kan utvecklarnas motstånd vara stort om de är rädda att förändringen kommer att kräva mera administrativt arbete och att de behöver lära sig ett nytt arbetssätt. Å andra sidan kan det vara yngre utvecklare som talar för förändringen.

Prepared by	Document
Mikko Koivunalho <mikko.koivunalho@iki.fi>	Jämförelse: Subversion och Git
Approved by	Date
	16 juni 2014

I 2012 tjeckiska *TMate Software* publicerade sitt *SubGit*-verktyg.(11) SubGit integrerar Git-arkiv och Subversion-arkiv tillsammans så att de bägge är speglar av varandra. Man kan använda både arkiv med dess egna klient-mjukvara. På det sätt alla användare kan välja vilken arkiv och vilka verktyg de vill använda.

Prepared by	Document
Mikko Koivunalho <mikko.koivunalho@iki.fi>	Jämförelse: Subversion och Git
Approved by	Date
	16 juni 2014

Referenser

[1] GitSvnComparison, https://git.wiki.kernel.org/index.php/GitSvnComparison.

[2] Spolsky, Joel, *Distributed Version Control is here to stay, baby*. Joel on Software. Retrieved 2014-06-13, http://joelonsoftware.com/items/2010/03/17.html

[3] *Distributed revision control*. Retrieved 2014-06-13, http://en.wikipedia.org/wiki/Distributed_version_control_system

[4] *Git*. Retrieved 2014-06-13, http://en.wikipedia.org/wiki/Git_(software)

[5] *Junio Hamano*. Retrieved 2014-06-13, http://en.wikipedia.org/wiki/Junio_Hamano

[6] *Subversion*. Retrieved 2014-06-15, http://en.wikipedia.org/wiki/Subversion_(software)

[7] Collins-Sussman, Ben. *Programmer Insecurity*. IBanjo pickin', grinnin', and pushin' bits. Published 12 June, 2008. Retrieved 2014-06-13, http://blog.red-bean.com/sussman/?p=96

[8] *Eclipse Community Survey Report 2013*. Retrieved 2014-06-13, http://www.slideshare.net/IanSkerrett/eclipse-survey-2013-report-final/14

[9] *ItJobsWatch*. Retrived 2014-06-13, http://www.itjobswatch.co.uk

[10] *Unix philosophy*. Retrieved 2014-06-14, http://en.wikipedia.org/wiki/Unix_philosophy,

[11] *SubGit, Safe Svn to Git Migration*, Retrieved 2014-06-15, http://subgit.com/index.html