



Blekinge Tekniska Högskola
Institutionen för Programvaruteknik och
Datavetenskap

Kandidatarbete i Datavetenskap, DVC001
C-nivå, 10 poäng
2003-05-28

Vad väljer din projektgrupp?

– Versionshanteringssystem för olika projekt

Författare: Anneli Sandberg
Ulrika Karlsson
Sara Olofsson

Handledare: Göran Gustafsson
Examinator: Guohua Bai



- Vad väljer din projektgrupp?
– Versionshanteringssystem för olika projekt.

Abstract

Title:

What does your project group choose? – Version control system for different project.

Authors:

Anneli Sandberg
Ulrika Karlsson
Sara Olofsson

Supervisor:

Göran Gustafsson

Area of problems:

Version control systems are developed for identification, to control changes, to guarantee that changes are implemented correctly and to report changes. Version control systems are used to save all history of different types of files that are used in the project group. For example, it can be used to check which dates the changes were made, by whom the files are modified and which comments are written about the changes.

We have chosen two version control systems: Windchill ProjectLink and CVS, Concurrent Versions System, to examine whether both size and benefit of the project control the choice of version control system.

Hypothesis and questions:

The thesis is based upon the hypothesis:” If the projects size and benefit of version control system are depending on each other then they are controlled by the choice of version control system.”

And our questions:

- Which size are the project, number of hours and persons?
- How many elements are used for each inner function?
- Which internal and external functions affect the choice of version control system?

Realization:

Contact where made with eight project groups that are using Windchill ProjectLink or CVS and conducted interviews with them. An investigation where made where we measured the elements on every version control system. Through the interviews and our own investigation we found the information so we could answer the hypothesis by the questions.

Conclusion:

The size and the benefit are not depending on each other so they are not controlled by the choice of version control system. Therefore we reject our hypothesis.



Vad väljer din projektgrupp?
– Versionshanteringssystem för olika projekt.

Sammanfattning

Titel:

Vad väljer din projektgrupp? – Versionshanteringssystem för olika projekt.

Författare:

Anneli Sandberg
Ulrika Karlsson
Sara Olofsson

Handledare:

Göran Gustafsson

Problemområde:

Versionshanteringssystem är utvecklade för att identifiera, kontrollera, rapportera och garantera att förändringarna har blivit rätt implementerade. Versionshantering används för att spara all historik av filer som man använder sig av i projektgrupper. Till exempel kan det användas till att se vilka datum ändringar gjorts, vem som har gjort ändringarna och vilka kommentarer som är skrivna om ändringarna.

Vi har valt att använda oss av två versionshanteringssystem, Windchill ProjectLink och CVS, Concurrent Versions System, för att undersöka om det är så att både projektets storlek och nytta styr valet av versionshanteringssystem.

Hypotes och frågeställningar:

Arbetet bygger på hypotesen: ”Om projektets storlek och nytta av versionshanteringssystem är beroende av varandra så styr det valet av versionshanteringssystem”

Och våra frågeställningar:

- Hur stort är projektet i antal timmar och personer?
- Hur många moment behövs för varje inre funktion?
- Vilka inre och yttre funktioner påverkar valet av versionshanteringssystem?

Genomförande:

Kontakt togs med åtta stycken projektgrupper som använder sig av Windchill ProjectLink eller CVS och intervjuer gjordes med dem. En egen undersökning har gjorts där vi mäter antal moment på varje funktion i versionshanteringssystemen. Genom intervjuer och vår egen undersökning fick vi reda på uppgifter som gör att vi kan besvara hypotesen med hjälp av frågeställningarna.

Slutsats:

Storleken och nyttan är inte beroende av varandra så att det styr valet av versionshanteringssystem. I och med detta stämmer inte vår hypotes.



- Vad väljer din projektgrupp?
- Versionshanteringssystem för olika projekt.

Förord

Vi vill tacka Anders Cronholm på CCA som har varit vår kontaktperson och vår främsta kontakt i detta arbete och som har hjälpt oss att hitta intervjupersoner för denna undersökning. Tack även till projektgrupperna som deltog i våra intervjuer.

Vi vill även tacka vår handledare Göran Gustafsson och vår examinator Guohua Bai på Blekinge Tekniska Högskola, för deras stöd och handledning.



- Vad väljer din projektgrupp?
– Versionshanteringssystem för olika projekt.

Innehållsförteckning

1. INLEDNING	1
1.1 DEFINITIONER	1
1.2 MÅL OCH SYFTE	1
1.3 HYPOTES OCH FRÅGESTÄLLNINGAR	2
1.3.1 <i>Avgränsningar</i>	2
1.4 MÅLGRUPP	2
2 BAKGRUND	3
2.1 PROJEKT	3
2.2 VERSIONSHANTERING I ETT PROJEKT	3
3 METOD	6
3.1 LITTERATUR.....	7
4 GENOMFÖRANDE	8
4.1. MOMENTUNDERSÖKNING.....	8
4.1.1 <i>Inhämtning av kunskap</i>	8
4.1.2 <i>Utförande</i>	8
4.2 INTERVJUER	9
4.2.1 <i>Intervjugrupper</i>	9
4.2.2 <i>Kontakt med intervjugrupper</i>	9
4.2.3 <i>Utförande</i>	10
4.2.4 <i>Bortfall</i>	10
5 TEORETISK GENOMGÅNG	11
5.1 IDENTIFIERING AV VERSIONER OCH RELEASER	11
5.1.1 <i>Vad är skillnaden på en version och en release?</i>	11
5.1.2 <i>Namnge versioner</i>	12
5.1.3 <i>Baslinje</i>	12
5.2 KONTROLLERA LAGRINGAR	13
5.2.1 <i>Fillåsning</i>	14
5.2.2 <i>Utan fillåsning</i>	14
5.3 EFFEKTIVT LAGRINGSSÄTT.....	14
5.3.1 <i>Branches</i>	14
5.3.2 <i>Merge</i>	15
5.3.3 <i>Delta</i>	15
5.4 ÄNDRINGSHISTORIK	16
5.5 NYA MÖJLIGHETER.....	16
5.6 CVS	17
5.6.1 <i>TortoiseCVS</i>	18
5.6.2 <i>WinCVS</i>	18
5.7 WINDCHILL PROJECTLINK	18
5.8 SKILLNAD MELLAN WINDCHILL PROJECTLINK OCH CVS	19



Vad väljer din projektgrupp?
– Versionshanteringssystem för olika projekt.

6 RESULTAT	21
6.1 UNDERSÖKNING	21
6.1.1 <i>Moment</i>	21
6.1.2 <i>Intervjuer</i>	22
6.2 RESULTAT AV UNDERSÖKNING.....	27
7 SLUTSATS	29
8 DISKUSSION	30
8.1 DISKUSSION KRING ARBETET	30
8.2 DISKUSSION KRING RESULTATET.....	30
8.3 FÖRSLAG PÅ FORTSÄTT ARBETE	31
9. KÄLLFÖRTECKNING	32
BÖCKER	32
INTERNET.....	32
MANUALER.....	33
INTERVJUER	33
10 ORDLISTA	34
BILAGA 1 MALL FÖR MOMENTUNDERSÖKNING	I
BILAGA 2 MOMENTUNDERSÖKNING TORTOISECVS	II
BILAGA 3 MOMENTUNDERSÖKNING WINCVS	III
BILAGA 4 MOMENTUNDERSÖKNING WINDCHILL PROJECTLINK	IV
BILAGA 5 MISSIV	VI
BILAGA 6 INTERVJUFRÅGOR	VII
BILAGA 7A INTERVJU BTH/KARLSKRONA KOMMUN	VIII
BILAGA 7B INTERVJU BTH/PRIVAT FÖRETAG	IX
BILAGA 7C INTERVJU BTH/IUC – OLOFSTRÖM	X
BILAGA 7D INTERVJU BTH/ERICSSON AB	XI
BILAGA 7E INTERVJU BTH/FÖRETAG	XII
BILAGA 7F INTERVJU BTH/UIQ	XIII
BILAGA 7G INTERVJU NN1	XIV
BILAGA 7H INTERVJU NN2	XV



- Vad väljer din projektgrupp?
- Versionshanteringssystem för olika projekt.

1. Inledning

I detta avsnitt beskrivs arbetets innehåll och inriktning på ämnet.

Om man använder ett versionshanteringssystem i en projektgrupp kan detta underlätta arbetet. I och med att ett versionshanteringssystem gör att man lätt kan hålla reda på vilken som är den senaste versionen, och alla filer är samlade på en plats där alla kan komma åt dem.

Versionshantering är en av de viktigaste bitarna i professionell systemutveckling. Att hantera all kod och dokumentation med största omsorg borde vara en självklarhet, men tyvärr så görs inte alltid detta. Idag finns det systemutvecklingsprojekt som nöjer sig med enkla hopsättningar av källkoden. Det förekommer också att det finns versionshanteringssystem tillgängliga för projektgruppen, men att de inte alltid används på bästa sätt.¹

Vi har valt att använda oss av två versionshanteringssystem, Windchill ProjectLink och CVS för att undersöka om det är så att både storleken på projektet och nyttan styr valet av versionshanteringssystem.

1.1 Definitioner

I detta arbete definieras storleken på ett projekt i antal timmar och personer.

Nyttan är ett mätvärde i antal moment, det vill säga vilken arbetskostnad det blir för att använda systemet och vilka inre och yttre funktioner som styr valet av versionshanteringssystem.

En inre funktion är en uppgift som finns i versionshanteringssystemet som man kan använda sig av. Till exempel att man kan ladda upp en fil till servern.

En yttre funktion är det som påverkar versionshanteringssystemet utifrån, alltså något som inte finns i systemet. Till exempel att ett versionshanteringssystem är webbaserat.

Ett stort projekt anser vi är ett projekt med minst 10 personer och är minst 8000 timmar, för alla personer som är inblandade i projektet. Detta har vi räknat ut genom att ta följande: ett projekt som är 20 veckor * 40 timmar per vecka * 10 personer. Varför vi har valt att ta dessa siffror beror på att vi utgår ifrån kursen Stort Programvaruprojekt på Blekinge Tekniska Högskola. Litet projekt är allt som är mindre än denna siffra.

1.2 Mål och syfte

Detta är ett intressant ämne för att det är viktigt med versionshanteringssystem inom projekt. Vi vill med detta arbete uppmärksamma ämnet

¹ <http://www.autark.se/versionshantering.html>, 2003-04-05



Vad väljer din projektgrupp? – Versionshanteringssystem för olika projekt.

versionshantering. Genom de undersökningar som görs kommer vi att antingen att kunna bevisa hypotesen eller motbevisa den. Vi vill att våra undersökningar ska visa om valet av versionshanteringssystem har betydelse för projekt. Detta genom att vi tittar på projektets storlek och nytta av versionshanteringssystem. Det kan vara till hjälp för projekt som inte vet vilket versionshanteringssystem de ska välja.

1.3 Hypotes och frågeställningar

”Om projektets storlek och nytta av versionshanteringssystem är beroende av varandra så styr det valet av versionshanteringssystem”

Vi vill närmare utreda om ovanstående hypotes är sann med hjälp av följande frågesällning:

- Hur stort är projektet i antal timmar och personer?
- Hur många moment behövs för varje inre funktion?
- Vilka inre och yttre funktioner påverkar valet av versionshanteringssystem?

Definition av projektets storlek, nytta och inre och yttre funktioner beskrivs i 1.1 Definitioner.

1.3.1 Avgränsningar

Vårt arbete kommer att avgränsas till att endast titta på projektgrupper som använder sig av versionshanteringssystem. Vi kommer inte att gå in på alla versionshanteringssystem utan avgränsa oss till de versionshanteringssystem som de projektgrupper som vi är i kontakt med använder. Och de versionshanteringssystem som projektgrupperna använder är CVS och Windchill ProjectLink. Behandlingen av de två versionshanteringssystemen kommer att vara ytlig, och hur versionshanteringssystemen fungerar tekniskt tas inte upp i detta arbete.

1.4 Målgrupp

Denna utredning vänder sig till intressenter som har eller vill ha kunskap om versionshantering. Detta kan innefatta projektgrupper som skall använda sig eller använder sig av någon sorts versionshanteringssystem, till exempel inom systemutveckling. Det bör betonas att grundläggande kunskaper behövs inom datavetenskap.



2 Bakgrund

Här nedan följer en introduktion till ämnet versionshantering; vad ett projekt är och varför versionshantering skall användas i projekt.

Systemutveckling är en process som handlar om att få användarnas behov i ett mjukvarusystem. Under utvecklingen handlar det inte bara om att göra rätt saker utan även att göra det på rätt sätt. Vid utvecklingen av ett system använder man sig av olika verktyg och metoder för att komma fram till en lösning.²

2.1 Projekt

När man arbetar med att utveckla ett system organiseras detta ofta som ett projekt som består av några systemutvecklare. Projekt används när man skall göra en helt ny uppgift som både är en engångsuppgift och som även kräver medverkan av människor med skilda ställningar och olika kompetenser. Ett projekt kännetecknas oftast även av att man skall uppnå ett visst resultat inom en viss tidsperiod. Vad man skall uppnå finns angivet, men målet kan vara något otydligt beskrivet och behöva förtydligast. I utgångsläget har man trots detta en bild av vad som skall utvecklas.³

I dagens samhälle blir projekten mer och mer komplexa, till storlek, finesser och tekniker som används. Detta gör att behovet av versionshantering är större än någonsin. Många system kommer ut till miljoner av användare, och kommer även ut i olika storlekar och former.⁴

2.2 Versionshantering i ett projekt

När projektgruppen bygger ett system uppkommer ständigt förändringar, därför är det viktigt att projekten kontrollerar dess effektivitet. Ifall projektet inte kontrollerar de förändringar som sker så kommer förändringarna att kontrollera projektet, och det är inte bra. Många okontrollerade förändringar kan göra ett väl fungerade system till kaos. Av denna anledning är versionshantering en viktig del av ett bra projekt.⁵

Versionshantering är aktiviteter som har till uppgift att kontrollera förändringar. Det blir då möjligt att upprätthålla förhållandet mellan förändringar. Versionshantering används bland annat för att spara historia av filer som projektgrupperna använder sig av. När man använder sig av versionshantering kan man se skillnader på de olika versionerna av filerna. Versionshantering sparar filerna på ett utrymmeseffektivare sätt, eftersom det bara är ändringarna som sparas.⁶

² Leon, 2000, s 13

³ Andersen, 1994, s 111

⁴ Leon, 2000, s 1

⁵ Pressman, 2001, s 225

⁶ Pressman, 2001, s 225



Vad väljer din projektgrupp? – Versionshanteringssystem för olika projekt.

Versionshanteringssystem är avgörande för framgång i ett rörligt projekt. En projektgrupp utan ett väl designat och lätt system gör att det blir mer extra jobb för varje utvecklare. I praktiken när man arbetar tillsammans kan man bygga en effektiv utvecklingsmiljö med hjälp av rätt system.⁷

Idag finns nu hundratals system som kan hjälpa till att få versionshanteringsprocessen att bli enklare.⁸

I ett versionshanteringssystem kan man hantera allt ifrån programkod till dokumentation, krav och buggrapporter.⁹

Idag finns det möjlighet att projektgrupper som jobbar flera tusen mil ifrån varandra kan arbeta som om de satt i samma kontor. Versionshanteringssystemen har numera tekniker som gör det möjligt att jobba på avstånd, parallellt och oberoende av varandra.¹⁰

Hur många personer som ingår i ett projekt kan ha en betydelse för hur versionshanteringen sker i ett projekt. Det kan vara projekt som endast består av en person eller projekt som innefattar flera personer. När endast en person gör ett projekt är det inte nödvändigt att använda ett versionshanteringssystem. Fördelarna med delad data och ständiga uppdateringsproblem försvinner. Men det är inte helt onödigt eftersom vårdslöshet, ouppmärksamhet och glömska kan göra att man skriver över sitt eget arbete.¹¹

Om en organisation har ett versionshanteringssystem och utvecklarna använder systemet, så har små projektgrupper fördelar med att kunna använda det. Stora projekt i organisationen där det ingår fler personer bör versionshanteringssystemet däremot användas.¹²

Effektiv versionshantering handlar inte bara om att ha ett versionshanteringssystem som kan hantera vem som gjorde förändringarna och när. Ett versionshanteringssystem skall vara skräddarsytt så att det passar varje projektgrupps karaktär, storleken på projektet, utvecklingsprocessen, kundens omfattning i projektet och så vidare. Det bästa sättet för att kolla hur versionshantering skall vara utformat för varje projekt är att ha en plan.¹³

⁷ Johanssen Hass, 2003, s 343

⁸ Leon, 2000, s 3

⁹ Pressman, 2001, s 226

¹⁰ Leon, 2000, s 75

¹¹ Leon, 2000, s 73-74

¹² Leon, 2000, s 73-74

¹³ Leon, 2000, s 7



Vad väljer din projektgrupp?
– Versionshanteringssystem för olika projekt.

Planen dokumenterar vilka aktiviteter som kan göras, hur de kan göras, vem som har ansvar för att göra en specifik aktivitet och vilka resurser som behövs.¹⁴

¹⁴ Leon, 2000, s 7



Vad väljer din projektgrupp?
– Versionshanteringssystem för olika projekt.

3 Metod

I metodavsnittet behandlas vilka tillvägagångssätt som vi ansåg lämpliga att använda för att samla in information och material. Vi motiverar hur vi har valt att göra vår undersökning.

Hypotesen testades genom att undersöka olika versionshanteringssystem, genom en egen momentundersökning och intervjuer.

Vi har även valt att göra en egen momentundersökning, om hur många moment det behövs för att utföra en viss funktion i versionshanteringssystemen. För att mäta nyttan av vilken arbetskostnad det blir att använda systemet har vi delat in versionshanteringssystemen i olika funktioner. Funktionerna valdes ut genom våra erfarenheter, genom läst litteratur om versionshanteringssystem och även genom utbildningstillfällen vi fått. De räknades sedan i antal moment för varje funktion, varje moment räknade vi som ett klick. En egen momentundersökning valdes att göras för att få en rättvis bild av hur många moment som behövdes för varje funktion.

Intervjuer valdes att göras med olika projektgrupper, för att få fram deras storlek på projektet och vilken nytta de har av versionshanteringssystemet, vilka funktioner som de anser nödvändiga. För att öka tillförlitligheten gjordes intervjuer då man får en direktkontakt med den personen man intervjuar.

Våra intervjuer är kvalitativ och inte kvantitativ. Detta beror på att vi inte bara är ute efter att få reda på en allmän uppfattning om versionshanteringssystemen utan även få en djupare kunskap om hur projektgrupperna upplever versionshanteringssystemet. I en kvalitativ intervju försöker man förstå och analysera helheten av intervjun. Vi arbetar med textmaterial och skrev ut alla intervjuer i sin helhet. Detta gjordes direkt efter intervjuerna medan det fortfarande fanns färskt i minnet. Efter varje intervju analyserade vi det insamlade materialet.¹⁵

För att testa denna metod med en intervju valdes att först göra en pilotstudie på en representativ grupp. Detta för att vi skulle få möjlighet att se om frågorna var tydliga, och se att vi fått med det mest väsentliga. För vår egen del gjordes denna pilotstudie för att träna oss själva på hur vi skulle klara av att anteckna svaren från intervjuerna, då vi under intervjuerna inte gjorde några ljudupptagningar.¹⁶

Vi har valt att göra intervjuer som har ganska låg grad av standardisering och strukturering, det vill säga att frågorna skickades ut innan, och därmed är frågornas ordningsföljd bestämd. Mycket få frågor tillkom under intervjuernas gång, detta för att göra det enklare för projektgrupperna att kunna delta i vår

¹⁵ Patel & Davidson, 1994, s 99-102

¹⁶ Patel & Davidson, 1994, s 68-70



Vad väljer din projektgrupp? – Versionshanteringssystem för olika projekt.

intervju och för att vi skulle kunna få bättre och mer exakta svar på de frågor som ställdes.¹⁷

Resultaten från intervjuerna, vår egen undersökning och den litteratur vi läst, hjälpte oss fram till ett resultat.

3.1 Litteratur

Den mesta information fick vi ifrån böcker men det inhämtades även en del information ifrån Internet. Böckerna valdes ut efter titeln, och innehållets relevans till ämnet. Litteraturen lästes igenom noga och de mest innehållsrika valdes ut. Bibliotekarie bokades för hjälp med att söka efter litteratur och artiklar. Liknande undersökningar inom ämnet har inte gått att finna.

¹⁷ Patel & Davidson, 1994, s 60-64



Vad väljer din projektgrupp?
– Versionshanteringssystem för olika projekt.

4 Genomförande

Här beskrivs hur vi gick tillväga med vår undersökning som består av; momentundersökning och intervjuer.

Vi valde att genomföra våra undersökningar på två versionshanteringssystem, Windchill ProjectLink och CVS. Windchill ProjectLink valdes för att vi kom i kontakt med en anställd på CCA, Collaborative Commerce Arena. Han hjälpte oss få fram projektgrupper som använde detta versionshanteringssystem. CCA är en enhet som drivs av Blekinge Tekniska Högskola i samarbete med olika företagspartner. CVS valdes för att de projektgrupper som vi kom i kontakt med använde sig av just detta versionshanteringssystem. Det var lättare att hitta projektgrupper inom högskolan än företag som använder CVS. Det är företaget PTC¹⁸ som tillhandahåller Windchill ProjectLink och CVS tillhandahålls av Cyclic Software¹⁹.

4.1. Momentundersökning

4.1.1 Inhämtning av kunskap

Det första vi gjorde var att inhämta kunskap om de båda versionshanteringssystemen. Detta gjordes genom att läsa igenom litteratur, där den mesta litteraturen bestod av manualer som tillhandahölls av de personer som introducerade oss i systemen. En del litteratur bestod även av Internet sidor.

Information om Windchill ProjectLink fick vi genom deltagande i ett par utbildningstillfällen som tillhandahölls av CCA. Vi fick även ett erbjudande av CCA att själva använda oss av Windchill ProjectLink under vårt arbete. Vi valde att utnyttja denna möjlighet för att få erfarenhet av hur ett webbaserat versionshanteringssystem fungerar.

Vi fick information om CVS genom deltagande i ett utbildningstillfälle som tillhandahölls av Konfigurations- och CVS-ansvarig i ett projekt på Blekinge Tekniska Högskola.

Ytterligare information om CVS gavs av en student på Blekinge Tekniska Högskola som använder detta versionshanteringssystem.

4.1.2 Utförande

På de versionshanteringssystem som vi valt att använda, letade vi efter subversioner, grafiskt användargränssnitt, till CVS. Sökning gjordes efter subversioner till CVS på ett par sökverktyg som finns på Internet. Det resulterade i att TortoiseCVS visades som en bland de första subversionerna. När vi sedan läste på sidan om TortoiseCVS så kom det fram att denna subversion är baserad på WinCVS. WinCVS är också en

¹⁸ <http://www.ptc.com>, 2003-03-10

¹⁹ <http://www.cyclic.com>, 2003-04-26



Vad väljer din projektgrupp? – Versionshanteringssystem för olika projekt.

subversion till CVS och då valdes även detta. Valet för att använda dessa två subversioner är för att Windchill ProjectLink är grafiskt och på så sätt blir det en mer rättvis bild vid jämförelse av de båda versionshanteringssystemen.²⁰

Med kunskaper genom erfarenhet, litteratur och utbildningstillfällena gjordes en mall (se Bilaga 1) med punkter för hur momenten skulle utföras.

Vi mäter sedan antal moment på de olika funktionerna;

- checka in en fil
- checka ut en eller flera filer
- ladda upp filer till servern
- se historik på en viss fil.

Dessa funktioner benämns med lite olika ord beroende på vilket versionshanteringssystem det handlar om.

Vi har sedan själva suttit var och en för sig, gjort alla punkterna och räknat ut hur många moment det behövs för varje funktion. Efter det satte vi oss och tittade på hur tillvägagångssättet var för varje funktion. Hur många moment som behövdes för varje funktion och utifrån detta sammanställdes antalet moment som behövdes. Detta gjorde vi på varje versionshanteringssystem för att få fram en rättvis bild. Varje moment räknade vi som ett klick, detta för att det blev lätt att jämföra de båda versionshanteringssystemen då ett av dem är webbaserat.

4.2 Intervjuer

Intervjuer har valts för att få fram storleken på projektgrupperna och vilka funktioner som anses nödvändiga, vilken nytta de har av versionshanteringssystemet.

4.2.1 Intervjugrupper

Begränsning har gjorts till åtta stycken projektgrupper som samtliga använder ett av de versionshanteringssystem som undersöks. Eftersom vi gör en kvalitativ undersökning har vi begränsat antalet projektgrupper. Projektgrupperna tilläts vara anonyma om de ville det.

4.2.2 Kontakt med intervjugrupper

Projektgrupperna som använder Windchill ProjectLink fick vi via vår kontakt på CCA. Han tog först kontakt med dem för att höra om projektgrupperna ville ställa upp på att vara med i vår undersökning. När vår kontakt på CCA fått klartecken från projektgrupperna tog vi själva kontakt med dem för närmare information om vad vår undersökning gick ut på.

²⁰ <http://www.tortoise cvs.org/>, 2003-04-28



Vad väljer din projektgrupp? – Versionshanteringssystem för olika projekt.

De flesta projektgrupperna som använder CVS kom vi i kontakt med genom kursansvarig för Stort programvaruprojekt på Blekinge Tekniska Högskola. E-post skickades till den som var kursansvarig för kursen och den personen vidarebefordrade vårt brev till dem som var projektledare i varje projekt. Efter det så tog projektledarna själva kontakt med oss. Resten av våra projektgrupper som använde CVS skedde genom personliga bekanta.

Vi har även under hela arbetets gång haft kontakt med alla projektgrupper via e-post och telefon för korta frågor.

4.2.3 Utförande

Efter våra kontakter via e-post och telefon så togs ytterligare kontakt med dem via e-post för att tala om vilken sorts undersökning vi tänkt göra, och i vilken omfattning. Frågorna (se Bilaga 6) som vi planerade att ställa under intervjun skickades ut 1-2 veckor innan. De fick då även ett missiv (se Bilaga 5), som innehöll information om varför vi gjorde denna intervju och varför den var viktig för oss. I vårt missiv talade vi också om att deras deltagande i intervjun får vara anonymt om de så önskade och vi skrev även vem de kunde kontakta om det skulle vara några frågor.²¹

De flesta av våra intervjuer som gjordes var genom personlig kontakt. I vår intervju presenterade vi oss själva, talade om syftet med vår intervju och att de hade en möjlighet att vara anonyma om de så önskade. De personer som intervjuades valdes ut av projektgrupperna själva. Varje intervju tog mellan tjugo och tjugofem minuter att genomföra.²²

Platserna för våra intervjuer var i sju av åtta fall i ett grupprum på Blekinge Tekniska Högskola eller i ett grupprum på företaget där projektgruppen arbetade. En intervju var tvungen att ske via telefon på grund av att det inte fanns någon möjlighet att ta sig dit.

Jämförelser mellan projektgruppernas storlek och nyttan gjordes för att komma fram till om de hade något samband med det versionshanterings-system projektgrupper har valt att använda.

4.2.4 Bortfall

Det blev inget bortfall i våra intervjuer eftersom vi haft god kontakt med de projektgrupper som intervjuades och uppdaterade dem under stora delar av arbetets gång. Alla projektgrupper sade under intervjuerna att vi gärna fick ta kontakt med dem efteråt om det skulle dyka upp fler frågor.

²¹ Patel & Davidson, 1994, s 60-64

²² Patel & Davidson, 1994, s 60-64



5 Teoretisk genomgång

Detta avsnitt ger mer kunskap om ämnet versionshantering.

Versionshantering har sitt ursprung ifrån försvarsindustrin. Produkterna som utvecklades var mycket små och finesserna på produkterna var dåliga. Men när produkterna blev mer komplexa var det omöjligt för en person eller grupp att behålla kontroll över design och produktion. När flera personer hade kontroll gjorde detta att information försvann, på grund av att det inte fanns några metoder för att dokumentera design och förändringarna som gjordes.²³

I början av 60-talet kom American Air Force på att de skulle göra en standard för versionshantering. Sedan när datorn blev populär så ökade utvecklingen av komplexa system och projektgrupperna blev större och större. Det blev nu ännu svårare för en person att sköta kontrollen så den amerikanska försvarsindustrin och många andra organisationer börja reda ut problemen med versionshantering. De kom alla ut med standarder för versionshantering.²⁴

Revision Control System (RSC) är en av de enklaste, äldsta och kanske det mest framgångsrika versionshanteringssystemet. RSC kan användas i de flesta Unix-miljöer och finns tillgängliga för många andra operativsystem. Flera moderna och fullt demonstrerade versionshanteringssystem bygger på idéer ifrån RSC.²⁵

Det finns olika versionshanteringssystem som tillåter en utvecklare att se olika skillnader mellan två filer. Vissa är textbaserade medan andra använder ett grafiskt gränssnitt (Graphical User Interface, GUI). Det kan vara en fördel att använda sig av olika versionshanteringssystem beroende på vad man skall utföra.²⁶

5.1 Identifiering av versioner och releaser

Att kunna identifiera versioner och releaser är viktig del av versionshantering.²⁷

5.1.1 Vad är skillnaden på en version och en release?

En version är ett exemplar av systemet som skiljer i de flesta fall ifrån tidigare version. Nya versioner av systemet kan ha olika funktioner, kapacitet eller bara reparera fel. Ifall det endast är små skillnader mellan versionerna, kallas de ibland varianter av varandra.²⁸

²³ Leon, 2000, s 3

²⁴ Leon, 2000, s 3

²⁵ Estublier, 1999, s 17-22

²⁶ Estublier, 1999, s 17-22

²⁷ Leon, 2000, s 91

²⁸ Sommerville, 2000 s 650



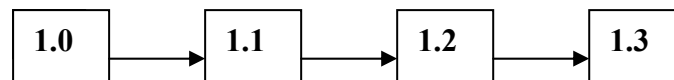
Vad väljer din projektgrupp? – Versionshanteringssystem för olika projekt.

Release är en version som blir släppt till kund. Varje systems release skall antingen inkludera nya funktioner eller vara lämplig för en annan hårdvaruplattform. Det finns alltid många fler versioner av ett system än releaser. Eftersom det finns versioner som endast skapas inom en organisation för intern utveckling eller testning som aldrig släpps till kunden.²⁹

5.1.2 Namnge versioner

Versionerna hanteras på ett enkelt sätt i ett schema. Schemat håller reda på versionerna och det är viktigt att ge dem en unik beteckning för att kunna se skillnader. Detta är extra viktigt i stora projekt där det finns hundratals filer och varje fil har ett antal versioner. Därför finns olika sätt att namnge versioner:

- **Versionsnummer:** Den första versionen kan man kalla 1.0, och en version av den kallas 1.1 och så vidare. I vissa steg, en ny release är skapad (release 2.0) och processen börjar då igen på version 2.1, 2.2 osv. Detta gör att man kan få tag i den önskade versionen. Detta är det vanligaste sättet att namnge versioner.



Figur 1 Linjär baserat

- **Attributbaserade namn:** Ett problem med tydliga versionsnamn är att de inte visar de olika attributen som används för att identifiera versioner. Därför kan man istället namnge dem efter kund, utvecklingspråk, utvecklingsstatus, hårdvaruplattform, vilket datum den skapades och av vem.
- **Förändringsorienterade namn:** Varje system har ett namn som är attributbaserat, men de är även associerade med en eller fler förändringar. Versionerna kan därför identifieras genom både namn och förändring.³⁰

5.1.3 Baslinje

Baslinjen spelar en viktig roll vid förändringar under utvecklingen. Under systemutvecklingsprocessen skapas en mängd filer. Till exempel dokument skapas, program kodas och testas och användarmanualer

²⁹ Sommerville, 2000 s 650

³⁰ Sommerville, 2000 s 650-651



Vad väljer din projektgrupp? – Versionshanteringssystem för olika projekt.

skrivs. Så fort en fil är klar tar versionshanteringssystemet hand om den.³¹

En baslinje är ett koncept som hjälper oss att kontrollera förändringarna utan allvarliga ändringar. Baslinjen hjälper till att hålla reda på fortsatta förändringar och är en milstolpe i utvecklingen. En fil blir först dokumenterad och granskad och om man finner fel så rättas de till. Filen blir sedan en baslinje och en grund för kommande förändringar.³²

Därefter blir den en bas för kommande utveckling och kan endast ändras genom formella förändringsmetoder. Baslinjer skall etableras i ett tidigt skede i ett projekt. Men att välja dem för tidigt gör bara att man för det mesta inte hittar den bästa lösningen, vilket visar sig när man testar systemet. När en fil blir en baslinje finns det inga fasta regler för. Så länge som en systemutvecklare arbetar med individuella moduler med lite påverkan, så behövs inte en baslinje. Däremot är det viktigare när man sätter samman filerna.³³

5.2 Kontrollera lagringar

Förändringar är ett faktum i systemutvecklingen. Kunderna vill göra ändringar på systemen, utvecklarna vill ändra tekniska bitar och företaget vill ändra strategin i projektet.³⁴

Kontroll över förändringar startar redan vid första dagen i ett projekt. En dialog mellan användaren och kunden skapar ett begär på ständiga förändringar. Dessa begär kan vara helt nya funktioner, förändringar eller tillägg i det redan implementerade systemet. Utvecklarna uppskattar varje begär som nya funktioner. Kunden prioriterar vilket som skall vara framtida arbete till nästa upprepning. Kunden driver prioriteringen och behoven för förändringarna medan utvecklaren tar emot dessa behov och fokuserar så mycket som möjligt i varje upprepning på detta.³⁵

Kontroll över förändringar får man med hjälp av att checka in och checka ut processerna. Detta sker vid två kontroller; accesskontroll och synkroniseringskontroll. Accesskontrollen undersöker om den person som försöker checka ut filen har tillåtelse att ändra i just den filen. Synkroniseringskontrollen hjälper oss att undersöka så att två personer inte skriver över varandra. Accesskontrollen kontrollerar att personen har rättigheter att checka ut och synkroniseringskontrollen låser filerna i en databas så att inga uppdateringar kan göras innan den som just nu checkade ut har ersatt versionen. Andra kopior kan bli utcheckade men andra uppdateringar kan inte bli gjorda. En kopia av en baslinje kallas ”extracted

³¹ Leon, 2000, s 58

³² Pressman, 2001, s 227

³³ Leon, 2000, s 58-60

³⁴ Pressman, 2001, s 227

³⁵ Johanssen Hass, 2003, s 345-346



Vad väljer din projektgrupp? – Versionshanteringssystem för olika projekt.

version” och kan ändras av utvecklarna. Efter lämpliga ändringar och tester av den ändrade versionen checkas filen in och en ny baslinje skapas.³⁶

5.2.1 Fillåsning

Innan en fil kan ändras måste den checkas ut med låsning. När man checkar ut filen blir den låst för alla andra, om det inte är tillåtet för flera personer att checka ut filen. För de som bara vill titta på filen kan de checka ut den utan låsning. Om man glömmer att checka in filen när man är klar så kan ingen annan använda filen.³⁷

5.2.2 Utan fillåsning

Vissa versionshanteringssystem tillåter att mer än en person kan checka ut och ändra i samma fil. Man kan sedan kombinera ihop filerna som blivit utcheckade, om de inte blivit ändrade på samma ställe. Versionshanteringssystemet måste kunna jämföra filer och meddela dessa till utvecklarna. (Mer om detta i avsnitt 4.3 Effektivt lagringssätt.)³⁸

En betydande fördel med ett versionshanteringssystem utan fillåsning är att utvecklaren kan arbeta samtidigt med samma filer. Utvecklaren kan till exempel arbeta med den senaste versionen medan en annan fixar ett fel i en gammal release eller version. Det gör att det blir möjligt för testgruppen att testa den senaste versionen vid samma tidpunkt som gruppen arbetar med den samma versionen.³⁹

Källkoden kontrollerar systemets förändringar och kan förbättra kapaciteten, vilket görs enkelt genom att checka in och checka ut. Förändringar tillkännages genom e-post mellan utvecklarna eller ord från mun till mun när gruppen sitter tillsammans.⁴⁰

5.3 Effektivt lagringssätt

Filer som checkas in och ut lagras på utvecklarens lokala hårddisk. Detta för att kunna hindra projektmedlemmar att ingripa och jobba med någon annans filer.⁴¹

5.3.1 Branches

Utveckling sker genom att filer checkas ut, förändringar sker och filer checkas sedan in igen. Men det är inte alltid som det går efter denna linjära utveckling. Branches använder man tillfälligt och tillåter två eller fler personer att jobba med samma fil oberoende av varandra, kanske även för olika mål. Detta gör man genom att man skapar två oberoende

³⁶ Pressman, 2001, s 234

³⁷ Leon, 2000, s 60-61

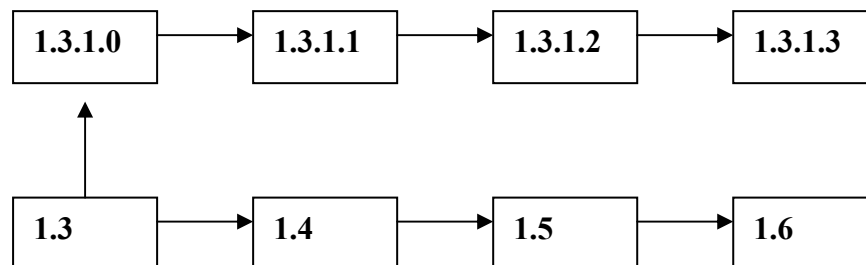
³⁸ Leon, 2000, s 60-61

³⁹ Asklund, 2002, s 38

⁴⁰ Johanssen Hass, 2003, s 347

⁴¹ Johanssen Hass, 2003, s 344-345

kopior. Branches skapas av många olika anledningar. De viktigaste är permanent branch vilken ändrar filen på olika sätt under begäran från olika operativsystem.⁴²



Figur 2 Grenar-branches

5.3.2 Merge

Merge betyder att man slår samman filerna och detta innebär att de ändringar som är gjorda vid en branch sammanställs till en ny filversion. Ifall förändringarna är gjorda på olika ställen i filen, så blir det enkelt att slå ihop filerna. Men ifall två personer har ändrat i samma rad i filen, så sker det en kollision. Personen som slår ihop filerna får då ta ett beslut om vilken ändring de skall behålla och vilken de skall kasta.⁴³

Versionshanteringssystemen har en automatisk funktion som slår ihop filerna, användaren behöver bara bestämma vilken version de skall använda om en kollision skulle uppstå.⁴⁴

Merge gör man när det inte längre finns en anledning att arbeta med samma filer. För det mesta består merge av en serie av revisioner och ytterligare branches kan skapas ifrån den ursprungliga. Att underhålla varianter är väldigt svårt.⁴⁵

5.3.3 Delta

I vanliga fall skall alla förändringar som sker registreras och alla versioner skall behållas. Detta för att alla användare inte alltid använder den senaste versionen. Versionshanteringssystemet skall kunna ta fram detaljer av den senaste versionen men även tidigare versioner. Kopior av alla versioner skall finnas på servern. Men detta gör man inte på grund

⁴² Leon, 2000, s 62-64

⁴³ Leon, 2000, s 62-64

⁴⁴ Leon, 2000, s 62-64

⁴⁵ Leon, 2000, s 62-64



Vad väljer din projektgrupp? – Versionshanteringssystem för olika projekt.

av utrymmesbrist. Istället skapar man ett delta, skillnaden mellan den nya versionen och den förra versionen.⁴⁶

Det finns två olika sätt att lagra delta.

- **Forward delta**, där man förändrar den senaste versionen hela tiden. Man utgår hela tiden ifrån grundversionen. Vill man ha tag på en senare filversion måste alla ändringar som befinner sig mellan grundversionen och den sökta versionen läggas till grundversionen.
- **Reverse delta**, där lagras även de senaste ändringarna som gjorts. Man behåller en fullständig kopia av den senaste versionen, och lagrar förändringar som har gjorts i tidigare versioner. När förändringen lagrats försvinner föregående version och den senaste versionen lagras istället.

Detta gör att det går långsammare att öppna en gammal filversion som lagrats med forward delta än med reverse delta. Med reverse delta blir det på motsvarande sätt svårare att gå bakåt bland versionerna.⁴⁷

5.4 Ändringshistorik

Alla versioner sparas och man kan när som helst ta fram en gammal version.⁴⁸

Ändringshistoriken gör att man alltid kan få svar på följande frågor: Vad hände? Vem gjorde det? När hände det? Vad påverkades?⁴⁹

Varje gång det inträffar en ny händelse i versionshanteringssystemet, en ny fil tillkommer eller blir uppdaterad så håller versionshanteringssystemet reda på detta i en rapport. Denna placeras i en databas så att utvecklaren kan få fram information genom ett nyckelord. Detta spelar en viktig roll i lyckade projekt. När många personer är involverade har man annars ingen koll på vad de andra i gruppen gör. Två utvecklare kan försöka att ändra i samma del av en fil och en konflikt uppstår.⁵⁰

5.5 Nya möjligheter

Stora företag och organisationer har länge haft access till globala nätverk, men under de senaste åren har hastigheten av Internet stigit. Grupper av utvecklare kan idag arbeta över hela världen i utvecklingen av samma system. Från olika platser kan de ändra tusentals filer, och ibland samma filer. I stort sett är utvecklarna inte tvingade att vara geografiskt placerade,

⁴⁶ Leon, 2000, s 68-70

⁴⁷ Leon, 2000, s 68-70

⁴⁸ <http://www.autark.se/versionshantering.html>, 2003-04-05

⁴⁹ Pressman, 2001, s 237-38

⁵⁰ Pressman, 2001, s 237-38



Vad väljer din projektgrupp? – Versionshanteringssystem för olika projekt.

vara på samma plats. Detta gör att det skapas nya krav på verktygen och dess system, som ska kontrollera utvecklingen. Supporten för skötseln av utvecklingen är uppgifter som blir mer komplicerade när personalen inte är geografiskt placerad. Oftast är system uppdelade i moduler, som sedan distribueras av olika grupper.⁵¹

5.6 CVS

CVS (Concurrent Versions System) är ett system för versionshantering som kan användas av flera personer samtidigt. Filer lagras tillsammans med de gjorda ändringarna på en central plats (CVS-trädet). Användarna arbetar sedan med lokala arbetskopior av filerna.⁵²

CVS är ett beprövat versionshanteringssystem, som är populärt hos bland annat programmerare. Det kostar ingenting i inköp, utan finns gratis att ladda hem på CVS:s hemsidan. I grunden är det kommandoinriktat, men numera finns det också subversioner till CVS.⁵³

Varje person checkar ut arbetskopior från servern till den egna datorn, för att kunna arbeta. När ändringar har gjorts checkas filerna in igen. Andra personer i projektgruppen kan då uppdatera sina arbetsfiler för att ta del av förändringarna som gjorts.⁵⁴

Du kan spara varje version av dina filer som du skapat. Detta skulle ta mycket utrymme i anspråk, på ditt konto. CVS lagrar alla versioner av filerna i en fil genom att använda delta.⁵⁵

Varje fil har ett versionsnummer och detta ökas varje gång filen checkas in. Versionsnumret gör att man kan se skillnad på vilken revision det är. Detta är något som bör stå i alla filer. Det gör det också möjligt att ta fram gamla version av filer med hjälp av versionsnumret.⁵⁶

Vissa editorer, som GNU Emacs, försöker att se till att samma fil aldrig kan ändras av två personer vid precis samma tidpunkt. Ifall någon använder en annan editor, så fungerar inte säkerheten. CVS löser detta problem genom att isolera olika utvecklare ifrån varandra. De kan låta varje utvecklare arbeta i sina egna filer, och CVS slår sedan samman filerna när varje utvecklare är klar.⁵⁷

CVS lagrar alla förändringar i en loggfil. Konflikter kan uppstå när flera personer har ändrat i samma fil. CVS kommer då att försöka lägga samman förändringarna och det gör CVS för det mesta bra. Men ibland uppstår

⁵¹ Estublier, 1999, s 17-22

⁵² <http://www.sics.se/~nfi/marketspace/cvs.html>, 2003-04-15

⁵³ <http://www.autark.se/versionshantering.html>, 2003-04-05

⁵⁴ <http://www.sics.se/~nfi/marketspace/cvs.html>, 2003-04-15

⁵⁵ <http://www.cvshome.org/docs/manual/>, 2003-04-07

⁵⁶ <http://www.sics.se/~nfi/marketspace/cvs.html>, 2003-04-15

⁵⁷ <http://www.sics.se/~nfi/marketspace/cvs.html>, 2003-04-15



Vad väljer din projektgrupp? – Versionshanteringssystem för olika projekt.

konflikter som CVS inte klarar av att lösa själv som till exempel när flera personer ändrat på samma rad. Vilken rad ska väljas?⁵⁸

Om det uppstår en konflikt när man uppdaterar filerna så talar CVS om det och markerar tydligt var den finns.⁵⁹

Vid användning av en subversion så visas statusen med hjälp av olika färger och markeringar:

Grön: Filen eller katalogen finns i CVS:en.

Röd: Filen eller katalogen finns i CVS:en, men är ändrade lokalt.

Lila: Förändringarna i filen är i konflikt med andra förändringar.

Blått frågetecken: Filen eller katalogen finns ej i CVS:en.⁶⁰

5.6.1 TortoiseCVS

TortoiseCVS är en subversion som gör det möjligt att arbeta med filer med CVS direkt ifrån Windows Explorer. Denna subversion är gratis.⁶¹

Med TortoiseCVS kan du checka ut, uppdatera, checka in och se skillnader på filer genom att höger klicka på filer och kataloger inom Explorer. Du kan även se statusen på en fil med hjälp av de normala ikonerna i Explorer.⁶²

5.6.2 WinCVS

Till skillnad från TortoiseCVS så måste man i denna grafiska klient logga in för att kunna använda WinCVS. Detta gör att man arbetar i en annan miljö än Windows Explorer.⁶³

5.7 Windchill ProjectLink

Windchill ProjectLink är den första lösningen på ett versionshanteringssystem som skapat något nytt genom att man kan utveckla i ett webbaserat arbetsutrymme. Här finns alla funktioner som anses nödvändiga för att man skall kunna arbeta på helt olika platser. I och med detta blir det enklare att få snabbare och smartare systemutveckling, som ger lägre kostnader och bättre produkter.⁶⁴

Windchill ProjectLink är baserad på Windchills teknologi, en webbaserad plattform som använder sig av standardiserade webbt teknologier för att utföra funktioner som behövs för ett versionshanteringssystem.⁶⁵

⁵⁸ <http://www.sics.se/~nfi/marketspace/cvs.html>, 2003-04-15

⁵⁹ <http://www.sics.se/~nfi/marketspace/cvs.html>, 2003-04-15

⁶⁰ Sandgren, 2003, s 3

⁶¹ <http://www.tortoise cvs.org/>, 2003-04-28

⁶² <http://www.tortoise cvs.org/>, 2003-04-28

⁶³ <http://www.devguy.com/fp/cfgmgmt/cvs/startup/>, 2003-04-26

⁶⁴ http://www.ptc.com/products/datasheets/projectlink_brochure.pdf, 2003-04-28

⁶⁵ http://www.ptc.com/products/datasheets/projectlink_datasheet.pdf, 2003-04-28



Vad väljer din projektgrupp? – Versionshanteringssystem för olika projekt.

Windchill ProjectLink tillhandahåller en projektbaserad lösning för samarbete med design projekt. De tillåter användaren att skapa och dela information i ett projekt med sina gruppmedlemmar. Gruppen får tillgång till versionshanteringssystemet och kan där spara och dela dokument, länkar till viktiga webbsidor, eller annan information. I Windchill ProjectLink kan projektgruppen sätta upp ett schema, eller planering för projektet. Man kan även tilldela projektets delleranser till olika användare. Det går att skapa schema för möten och online diskussioner. Användarna kan även skicka ett e-postmeddelande till andra användare för att de ska granska eller godkänna ett dokument.⁶⁶

Att skapa ett nytt projekt i Windchill ProjectLink är enkelt; en användare loggar in, sätter upp ett projekt och användarna i projektet. Dessa användare kan sedan lätt gå in på Windchill ProjectLink och kolla hur projektet utvecklas.⁶⁷

Filerna lagras i kataloger och tillträde kontrolleras enligt en specifikation. Det finns en komplett ändringshistorik.⁶⁸

I Windchill ProjectLink kan man få en bild över varje användares specifika projekt och aktiviteter med hjälp av funktionen Home. En användare kan där enkelt ta hand om sina ansvarsuppgifter och leverera dem. Deltagarna kan utveckla, tillträda, och underhålla filer så rätt information är tillgänglig när en projektmedlem behöver den.⁶⁹

Tillträde till individuella dokument görs via kontrollregler, det vill säga att man är behörig. Man har även en personlig anteckningsbok för att hantera länkar till viktig information.⁷⁰

I Windchill ProjectLink visas en röd ”bock” när en fil är utcheckad och det betyder att någon arbetar med den. Windchill ProjectLink låser filerna när någon checkar ut dem. Detta för att undvika att två personer sitter och arbetar med samma fil samtidigt. Det går bara att checka ut det senaste dokumentet men det går att titta på äldre dokument.⁷¹

5.8 Skillnad mellan Windchill ProjectLink och CVS

De viktigaste skillnaderna är att Windchill ProjectLink är ett versionshanteringssystem som är webbaserat. CVS tillåter att två eller flera

⁶⁶ Windchill® ProjectLink™ User's Guide, 2002

⁶⁷ Windchill® ProjectLink™ User's Guide, 2002

⁶⁸ http://www.ptc.com/products/datasheets/projectlink_datasheet.pdf, 2003-04-28

⁶⁹ http://www.ptc.com/products/datasheets/projectlink_datasheet.pdf, 2003-04-28

⁷⁰ http://www.ptc.com/products/datasheets/projectlink_datasheet.pdf, 2003-04-28

⁷¹ Windchill® ProjectLink™ User's Guide, 2002



6 Resultat

Här beskrivs vår undersökning som bestod av; momentundersökning och intervjuer.

6.1 Undersökning

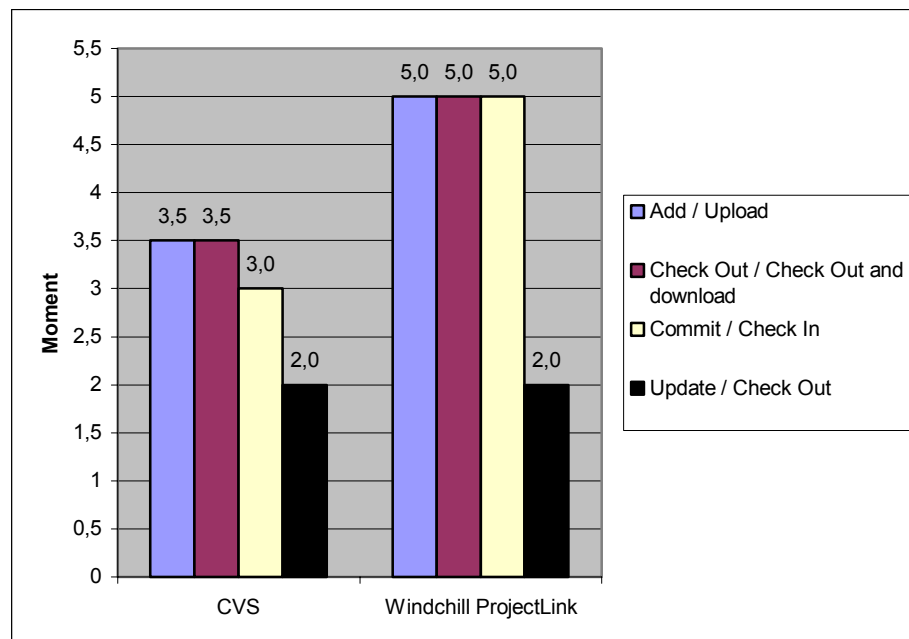
6.1.1 Moment

I och med att vi har tittat på två subversioner till CVS har vi tagit ett medelvärde på antal moment som behövs för en viss funktion.

Varje moment har räknats som ett klick, det vill säga varje gång som du klickar med höger eller vänster musknapp.

Utförligare beskrivning se Bilaga 2 Momentundersökning TortoiseCVS, Bilaga 3 Momentundersökning WinCVS och Bilaga 4 Momentundersökning Windchill ProjectLink.

Diagram 1 Antal moment per inre funktion



- **Add / Upload**
En ny fil görs som ska laddas upp till servern.
- **Check out / Check out and Download**
I CVS laddas alla filer som finns i katalogen ner, i Windchill laddas den filen som vi vill arbeta med ner.
- **Update / Check out**
I CVS så laddas alla ändringar som gjorts på filen ner, i



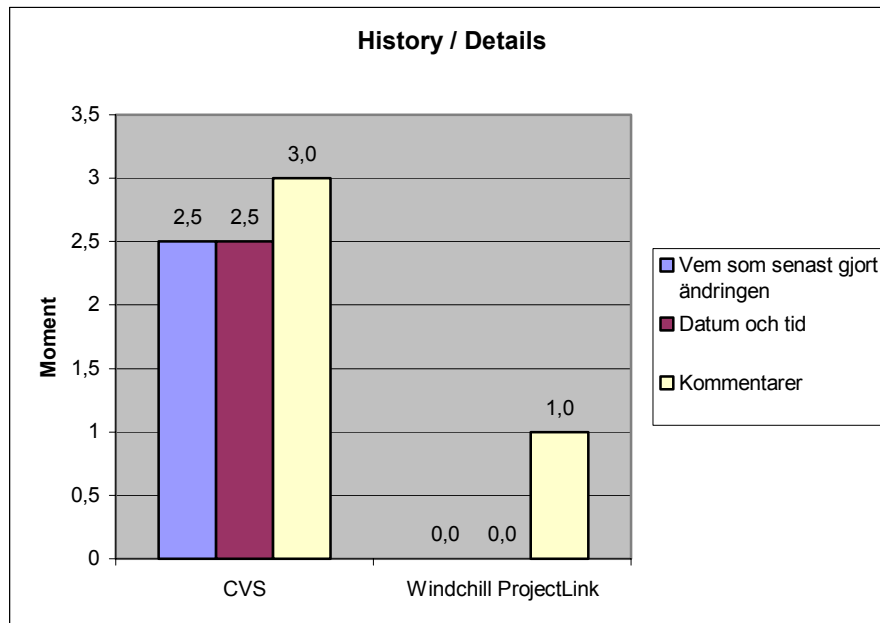
Vad väljer din projektgrupp? – Versionshanteringssystem för olika projekt.

Windchill finns filen redan tillgänglig på datorn och vi ska bara markera att någon arbetar med den.

- **Commit / Check in**

En fil som är gjord ändringar i laddas upp till servern med de nya ändringarna.

Diagram 2 Antal moment för historik



- **History / Details**

- **Se vem som gjort senaste ändringen**
Se vem, i projektgruppen, som gjort senaste ändringen i en fil.
- **Se vilket datum och vilken tid senaste ändringen är gjord**
Se vilket datum och vilken tid som senaste ändringen är gjord i en fil.
- **Se vilka kommentarer som är skrivna om senaste ändringen**
Se skrivna kommentar om senaste ändringen som är gjort i en fil.

Summan blir att för CVS gör man 20 moment för alla funktionerna och i Windchill ProjectLink gör man 18 moment för alla funktionerna.

6.1.2 Intervjuer

I detta kapitel sammanställs intervjuerna och i en del infogas i diagram för att göra det hela mer överskådligt.



Vad väljer din projektgrupp? – Versionshanteringssystem för olika projekt.

I intervjun ställdes frågor som handlade om projektens storlek och de olika yttre och inre funktionerna i versionshanteringssystemet.

Frågorna från intervjuerna står här nedan i fetstil innan sammanställningen av svaren.

Utförligare svar från intervjuerna se Bilaga 7A till Bilaga 7H .

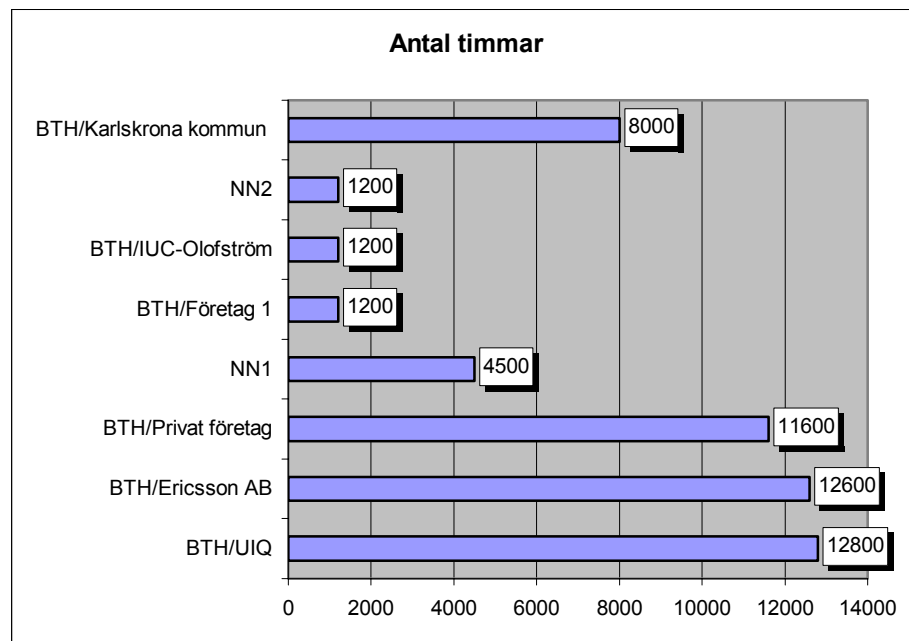
Vilket företag/skola tillhör Er projektgrupp?

Vi intervjuade representanter från åtta olika projektgrupper. Sex av projektgrupperna var på Blekinge Tekniska högskola, i samarbetade med företag. Därför har vi valt att benämna grupperna genom att använda oss av förkortningen på skolan och sedan det företag som de jobbar mot i deras projekt.

Två grupper har valt att vara anonyma och dessa har vi valt att kalla NN1 och NN2.

Hur många personer ingår i Er projektgrupp?

Diagram 3 Antal personer per projektgrupp

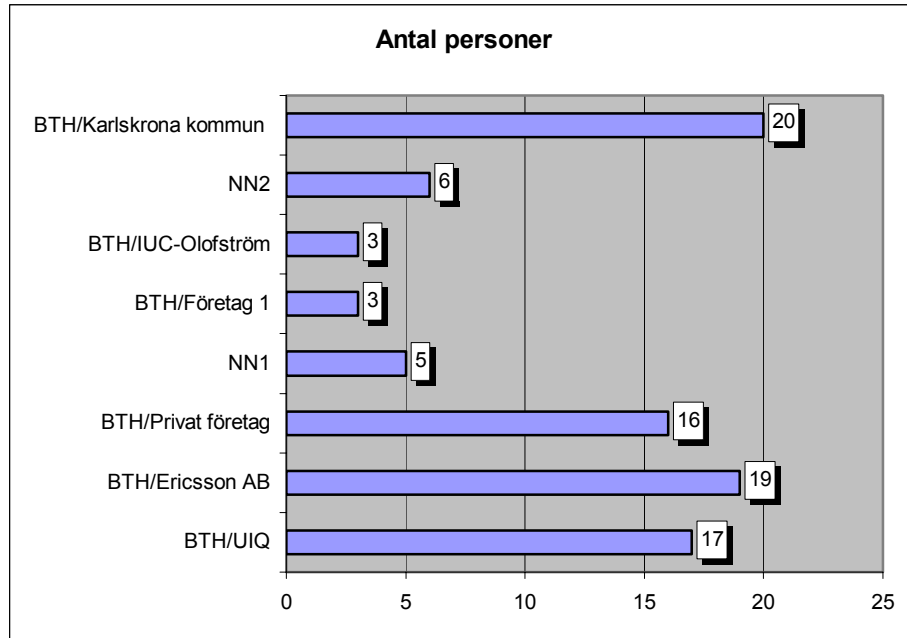




Vad väljer din projektgrupp? – Versionshanteringssystem för olika projekt.

Totalt, hur många timmar är projektet för hela projektgruppen?

Diagram 4 Antal timmar fördelade på de olika projektgrupperna



Vilket versionshanteringssystem använder Ni i projektgruppen?

BTH/UIQ använder sig av CVS på servern och subversionen TortoiseCVS på klientsidan. TortoiseCVS använder de endast för källkod. Till all dokumentation har de ett egenutvecklat program.

BTH/Ericsson AB använder WinCVS och IntelliJ IDEA på klientsidan. WinCVS använder de till dokument och konfigurationsfiler. IntelliJ IDEA använder de till källkoder.

BTH/Privat företag använder sig av CVS på servern och WinCVS som klient. WinCVS använder de till all källkod och dokumentation.

Versionshanteringssystemet som NN1 använder är CVS på serversidan och subversionen WinCVS på klientsidan, där de sparar källkod och dokumentation.

Resterande grupper, BTH/Företag, BTH/IUC-Olofström, NN2 och BTH/Karlskrona kommun, svarade att de använder Windchill ProjectLink.



Vad väljer din projektgrupp? – Versionshanteringssystem för olika projekt.

Varför använder Ni just detta system?

BTH/UIQ använder sig av CVS för att det är gratis och plattformsoberoende. De hade använt det tidigare och de tyckte att det är lätt att hitta information om systemet.

BTH/Ericsson AB har haft en föreläsning om CVS och därför visste de hur det fungerade. De tycker även att det är en bra standardlösning som också ger bra stöd i deras utvecklingsmiljö. De anser också att valet påverkades av att det är gratis.

BTH/Privat företag använder CVS för att det finns personer inom gruppen som använt det tidigare. De hade bara hört bra åsikter om CVS.

NN1 tycker att CVS är tillförlitligt, säkert och bra grafiskt. De menar att det är bra för att det är plattformsoberoende och gratis. Som server är det lämpligt att använda sig av något Unix-liknande operativsystem. De tycker att det är bra att man via SSH kan komma åt filerna på ett säkert sätt.

BTH/Företag fick erbjudande om att använda Windchill ProjectLink och tog chansen att göra det. De fick då en grundläggande genomgång och visste inget tidigare om ämnet. De tycker att Windchill ProjectLink är bra för att det är webbaserat. De sade som exempel att om en av dem åkte hem över en helg behövde de inte lägga allt på disketter.

BTH/IUC-Olofström blev tvingade att använda Windchill ProjectLink av deras kursansvarige.

NN2 blev tvingade att använda Windchill ProjectLink för att få kunskap om hur ett versionshanteringssystem fungerar.

BTH/Karlskrona kommun har använt Windchill ProjectLink i tidigare projekt och hade därför erfarenhet, dessutom anser de att det är bra att versionshanteringssystemet är webbaserat.

Vilka funktioner använder Ni i systemet?

BTH/UIQ använder sig av funktionerna Check out, Commit, Add, Remove och Update.

BTH/Ericsson AB anser sig använda de mest grundläggande funktionerna, det vill säga Check out, Commit, Add, Check in och Tag.

BTH/Privat företag använde sig av Check out, Update, Commit, Tag och Merge

NN1 svarade Check out, Commit, Remove, Update, Add, Tag, Diff, Status, Import, Admin och Init.



Vad väljer din projektgrupp? – Versionshanteringssystem för olika projekt.

BTH/Företag använder sig av Files för att lägga upp filer. De använder också Plan och Deliverables.

BTH/IUC-Olofström använder sig av funktionerna Plan och Files. De är inne och uppdaterar på Windchill ProjectLink cirka en gång i veckan.

BTH/Karlskrona kommun och NN2 använder Files, Plan, Deliverables, Home och Forum.

Vilka funktioner tycker ni är viktigast för det systemet Ni använder?

BTH/UIQ anser att det viktigaste är Commit och Update. Samt att de i CVS kan arbeta flera personer på med samma fil och när de kör Update kontrollerar systemet om det uppstod någon konflikt. De använder Check out för att testa de filer som finns på servern samt synkronisera dem.

BTH/Ericsson AB tycker att de viktigaste funktionerna är Check out, Commit, Add, Check in och Tag.

BTH/Privat företag anser att de viktigaste funktionerna är samma, som de använder, det vill säga; Check out, Update, Commit, Tag och Merge.

NN1 menar att Update är en av de viktigaste funktionerna då man kan lita på att den inte skriver över några filer man ändrat lokalt.

BTH/Företag säger att de tycker Files är den viktigaste funktionen, eftersom det är där de kan lägga upp sina filer.

BTH/IUC-Olofström bedömer att Plan är den funktion som de anser är viktigast, eftersom de ändå sparar sina filer på disketter. De får via Plan bra koll på vad som gjorts och inte gjorts i projektet.

NN2 och BTH/Karlskrona kommun tycker båda att det viktigaste är funktionerna i Files och att det är webbaserat. De menar också att webbaserade versionshanteringssystem har fördelar för att man alltid kan få tillgång till filerna. Men det skapar också problem om servern ligger nere.

Finns det några funktioner som Ni saknar/inte använder i Ert system?

BTH/UIQ använder sig inte av funktionerna Tag och Branch. De anser sig sakna funktionen Diff, men menar också att det finns andra subversioner som har den funktionen.

BTH/Ericsson AB skulle vilja använda Branch, men just nu anser de att det är för mycket buggar i systemet. De gör inte kommentarer när de



Vad väljer din projektgrupp? – Versionshanteringssystem för olika projekt.

uppdaterar filer och tycker att det är svårt att ta bort filer. De skulle föredra en funktion för att se alla senaste versioner på ett ställe. De tycker också att programmet känns osäkert och användbarheten på klientsidan inte är så bra. De tror säkert det finns många fler funktioner som de inte använder, men kom inte på några.

BTH/Privat företag tycker inte att de saknar något i versionshanteringssystemet. De anser att CVS är stort och innefattar många funktioner som de inte använder, detta har både fördelar och nackdelar.

NN1 anser att de säkert finns funktioner som de vill ha, men kommer inte på någon. De använder inte subversioner till CVS som till exempel CVS Bug Tracker och CVSWeb. De anser att det saknas tempo i versionshanteringssystemet, de vill att hastigheten i systemet ska ökas.

BTH/Företag anser sig inte använda Forum, Home eller e-postfunktionen i Windchill ProjectLink. De saknar dock en knapp för utloggning ur systemet, de tycker att det känns otillförlitligt att inte kunna logga ut, utan att stänga fönstret. De tycker att nackdelen med webbaserat är att servern ibland ligger nere.

BTH/IUC-Olofström anser att de inte saknar något, då de inte använt något versionshanteringssystem tidigare. De tyckte att inlärningsprocessen tog mycket tid i anspråk, men att det nu går bättre. De tycker att Windchill ProjectLink är för långsamt och efterlyser snabbhet. BTH/IUC-Olofström använder inte funktionerna Forum eller Home.

NN2 och BTH/Karlskrona kommun vet inte om några funktioner som de inte använder. De anser sig inte sakna något i versionshanteringssystemet, då de inte använt något annat.

6.2 Resultat av undersökning

För att kunna svara på vår frågeställning om hur många moment som behövs för varje inre funktion gjordes två momentundersökningar, en för CVS och en för Windchill ProjectLink. Resultaten av dessa momentundersökningar återfinns i avsnittet 6.1.1 Moment. För de inre funktionerna har vi kommit fram till att de som redovisas är de mest grundläggande. Detta har vi fått fram genom intervjuer som gjorts med representanter från projektgrupper och dels från våra egna erfarenheter. Våra egna erfarenheter har kommit från att vi själva använder Windchill ProjectLink och även erfarenheter från att vi har tittat på andra versionshanteringssystem. Det som konstateras efter den gjorda momentundersökningen är att i Windchill ProjectLink behövs det mindre moment för att genomföra de grundläggande inre funktionerna.



Vad väljer din projektgrupp? – Versionshanteringssystem för olika projekt.

När vi har gjort våra intervjuer har vi kommit fram till följande:

Av de grupper som använder CVS var tre stycken stora projekt och ett litet projekt.

I Windchill ProjectLink var det ett stort och tre litet. (definition se 1.1 Definitioner)

De inre funktionerna som projektgrupperna inom CVS använder är Commit, Update, Check out, Add och Tag.

Alla i projektgrupperna som använt Windchill ProjectLink använder funktionerna som finns under Files, det vill säga Check out, Upload, Check in och Details.

Samtliga projektgrupper använde även Plan och Deliverables för att strukturera sitt projekt. De inre funktionerna Forum och Home användes inte i lika stor utsträckning.

Inom CVS anser projektgrupperna att Commit, Update och Check out är de viktigaste inre funktionerna.

De inre funktionerna som projektgrupperna anser viktigast är inom Windchill ProjectLink, Check out, Update, Check in, Upload och Details.

Projektgrupperna som använder CVS saknar i de flesta fall inget i versionshanteringssystemet.

I Windchill ProjectLink saknar projektgrupperna ingenting på grund av att de flesta projektgrupperna inte använt något versionshanteringssystem tidigare.

Branch är en av funktionerna som de flesta projektgrupperna anser att de inte använder sig av i CVS.

I Windchill ProjectLink vet de inte om några funktioner som de inte använder.

I projektgrupperna som använder CVS så hade de tidigare erfarenhet av versionshanteringssystem. CVS är gratis och plattformsoberoende därför påverkade detta deras val av versionshanteringssystem.

Många av projektgrupperna som använder Windchill ProjectLink tycker det är bra för det är webbaserat. Vissa av grupperna använder Windchill ProjectLink för att de har blivit tvingade av kursansvariga, och för att de skulle lära sig ett versionshanteringssystem.



Vad väljer din projektgrupp?
– Versionshanteringssystem för olika projekt.

7 Slutsats

I detta avsnitt kommer vi att titta på det resultat som framkommit av vår egen undersökning och av intervjuer med projektgrupperna. Med dessa uppgifter besvaras hypotesen med hjälp av frågeställningarna.

Storleken på projektgruppen har inte med valet av versionshanteringssystemet att göra, eftersom vi i vår undersökning har kommit fram till att både stort och litet projekt använder sig av de båda versionshanteringssystemen.

Gällande nyttan så har det kommit fram att Windchill ProjectLink har mindre moment än CVS. Detta kom fram efter vår egen momentundersökning. Vi kan också säga att det inte finns någon väsentlig avvikelse eftersom det rör sig om små skillnader.

Under intervjuerna har det kommit fram vilka som är de viktigaste funktionerna, och vi ser det som att det inte heller finns någon skillnad. De flesta funktioner som uppges finns i de båda versionshanteringssystemen, och påverkar därför inte valet av versionshanteringssystem.

Det som dock påverkar valet av versionshanteringssystem är de yttre funktionerna. Bland de yttre funktioner som nämndes påverka valet av CVS är att det gratis och plattformsoberoende. Medan de yttre funktioner som påverkar valet av Windchill ProjectLink är att det är webbaserat och att intervjupersonerna inte hade någon tidigare erfarenhet av versionshantering.

Storleken och nyttan är alltså inte beroende av varandra så styr det valet av versionshanteringssystem.

Genom våra undersökningar har vi kommit fram till att vår hypotes inte stämmer.



8 Diskussion

I detta avsnitt kommer vi att ta upp saker som kan ha påverkat vårt arbete och resultat. Vad som inte har gått som väntat och vad som har gått enligt planerna. Efter det kommer vi att ge förslag på fortsatt arbete.

8.1 Diskussion kring arbetet

Under vårt arbete har vi till stor del upplevt att allt har fungerat bra. Under arbetets gång har vi fått bra kunskap om vad ett versionshanteringssystem är och vilka fördelar det för med sig.

När vi skulle ta kontakt med projektgrupper, som använde CVS och Windchill ProjectLink, var de svåra att hitta.

När vi väl fick tag i projektgrupper som ville vara med i en intervju gick de bra. Kontakt fanns hela tiden med projektgrupperna vilket gjorde att det inte blev några bortfall. Projektgrupperna ställde även upp med att komplettera intervjun i efterhand.

Under arbetets gång har vi använt Windchill ProjectLink, vilket inte bara har upplevts som positivt. Servern har varit nere många gånger vilket har gjort att vi inte kunnat komma åt våra dokument. Däremot har det underlättat arbetsgången vid hemarbete. Vi har också kunnat hålla reda på våra olika versioner. Nu i efterhand hade vi även valt att använda oss av CVS till arbetet, för att få mer erfarenhet av detta versionshanteringssystem.

Litteratur om ämnet var något som i början av arbetet var väldigt svårt att hitta. Det fanns inga böcker som endast behandlade versionshantering, utan information hittades istället i systemutvecklingsböcker.

Momentundersökningen tog längre tid än planerat på grund av att det tog längre tid att få en överblick av versionshanteringssystemen.

8.2 Diskussion kring resultatet

I början av arbetets gång hade vi en önskan att komma i kontakt med projektgrupper på företaget. Så blev inte fallet vilket kan ha påverkat vårt resultat.

Resultatet kan ha påverkats av att vi valt att undersöka just Windchill ProjectLink och CVS. Om vi hade behandlat fler versionshanteringssystem kan detta ha påverkat vårt resultat på ett annat sätt. En annan sak som kan ha betydelse är vår urvalsstorlek av projektgrupper. Ytterligare något som kan ha påverkat vårt resultat är att vi inte valde att titta på subversionen CVSWeb till CVS. Vi har inte tagit någon hänsyn till att Windchill ProjectLink inte är gratis och att projektgrupper som använder sig av detta versionshanteringssystem har support och garanti istället.



Vad väljer din projektgrupp? – Versionshanteringssystem för olika projekt.

I vår hypotes undersöks om både nyttan och projektets storlek är beroende av varandra vilket styr valet av versionshanteringssystem. Om vi istället skulle ha undersökt om nyttan eller projektets storlek så skulle hypotesen till viss del vara sann.

Under arbetets gång kom vi på att det är inte bara de funktioner som finns i systemet som påverkar varför man väljer ett visst versionshanteringssystem.

Vi har lagt märke till att Windchill ProjectLink inte lämpar sig för stora projektgrupper eftersom flera personer inte kan använda en fil samtidigt. I Windchill ProjectLink låser man en fil när man checkar ut den och detta anser vi inte är lämpligt för källkod. I CVS kan man däremot arbeta med samma fil samtidigt.

Intervjuerna med representanter från projektgrupperna kan ha påverkat vårt resultat. Dels vem som har svarat på frågorna och andra faktorer, så som stress och miljö.

Intervjupersonerna inom CVS uppfattades vara mer kunniga inom området. De hade valt CVS för att de visste om fördelarna och hur det fungerade.

I och med att vi valde att göra intervjuer så ökade tillförlitligheten och en direktkontakt med intervjupersonerna fanns. Det blev även lättare att förmedla vårt budskap.

8.3 Förslag på fortsatt arbete

Ett förslag som vi hade tänkt oss från början att arbeta med var att titta på skillnaden mellan versionshantering på projekt som är belägna på företag och högskolor.

Ett annat förslag kan vara att undersöka webbaserade versionshanteringssystem gentemot icke webbaserade.

Ett ytterligare alternativ är att undersöka om en viss subversion är mer lämpad för en viss projektgrupp.



Vad väljer din projektgrupp?
– Versionshanteringssystem för olika projekt.

9. Källförteckning

Böcker

Andersen, Erling S – Systemutveckling – principer, metoder och tekniker, 1994 Studentlitteratur, Lund.

Asklund, Ulf – Configuration Management for Distributed Development in a Integrated Environment, Doctoral dissertation, 2002

Estublier, Jacky (Volume Editor) – System configuration management, 1999 Springer-Verlag, Berlin

Jonassen Hass, Anne Mette – Configuration management principles and practice, 2003 Pearson Education, Inc.

Leon, Alexis – A Guide to Software Configuration Management, 2000 Artech House, Norwood.

Patel, Runa & Davidson, Bo – Forskningsmetodikens grunder, 1994 Studentlitteratur, Lund.

Pressman, Roger S – Software engineering, a practitioner's approach, 2001 R. R. Donnelley & Sons Company.

Sommerville, Ian – Software Engineering, 6th Edition, 2000

Internet

<http://www.autark.se/versionshantering.html>, 2003-04-05, Autark Datakonsult AB

<http://www.cvshome.org/docs/manual/>, 2003-04-07, CollabNet, Inc.

<http://www.cyclic.com>, 2003-04-26, CollabNet, Inc.

<http://www.devguy.com/fp/cfgmgmt/cvs/startup/>, 2003-04-26

<http://www.ptc.com>, 2003-03-10, Parametric Technology Corporation

http://www.ptc.com/products/datasheets/projectlink_brochure.pdf, 2003-04-28, Parametric Technology Corporation

http://www.ptc.com/products/datasheets/projectlink_datasheet.pdf, 2003-04-28, Parametric Technology Corporation

<http://www.sics.se/~nfi/marketspace/cvs.html>, 2003-04-15, Swedish Institute of Computer Science

<http://www.tortoise cvs.org/>, 2003-04-28



Vad väljer din projektgrupp?
– Versionshanteringsystem för olika projekt.

Manualer

Sandgren, Roy - Quill Instant messaging CVS handledning version 1.04,
2003-03-03

Windchill® ProjectLink™ User's Guide, Release 6.2.6, December 2002,
Parametric Technology Corporation'

Intervjuer

Dogertz, Julia & Lundin, Maria, studenter Blekinge Tekniska Högskola.
Examensarbete i samarbete med Företag 1.

Falkenberg, Gabriel & Andersson, Joakim, studenter Blekinge Tekniska
Högskola. Projektarbete i samarbete med Ericsson AB

Sandgren, Roy, student Blekinge Tekniska Högskola. Projektarbete i
samarbete med UIQ.

Svensson, Pär, student Blekinge Tekniska Högskola. Projektarbete i
samarbete med privat företag.

Wahlberg, Carl-Johan, student Blekinge Tekniska Högskola.
Examensarbete i samarbete med IUC-Olofström.



Vad väljer din projektgrupp?
– Versionshanteringssystem för olika projekt.

10 Ordlista

Add	Lägga till nya filer i CVS strukturen
Admin	Ofta en förkortning av Administratör.
Check in	Lägger in ett specifikt objekt på servern.
Check out	Hämta ut ett objekt från servern. (Windchill ProjectLink) Normalt används det till att beskriva den första återvinningen av en hel modul från ”själva databasen”. (CVS)
Commit	Föra in dina ändringar, av den specifika fil, i CVS:en.
CVS-Bugtracker	Är en webbaserat verktyg för att spåra fel i systemutvecklingen, lika väl som att rapportera småsaker och nya kännetecken för redan existerande filer.
CVSWeb	Ett webbaserat gränssnitt för CVS.
Deliverables	Varje milstolpe kan bestå av ett antal delmål. (Windchill ProjectLink)
Details	Visa upp det speciella för objektet som även inkluderar alla associerade markeringar, vad som är ändrat på objektet och all diskussion och relaterade listor till objektet. (Windchill ProjectLink)
Editor	Textredigerare.
Export	Refererar till en uttagning av en hel modul från “själva databasen”.
Files	Här sparas och visas de filer som är relaterade till ett projekt. (Windchill ProjectLink)
Forum	Ett diskussionsforum där användarna som deltar i ett projekt kan kommunicera med andra användare i samma projekt. (Windchill ProjectLink)
GNU Emacs	En textredigerare för en rad olika typer av filer, allt från ren text, källkod till HTML filer.
GUI	Graphical User Interface. Användargränssnitt.



Vad väljer din projektgrupp? – Versionshanteringssystem för olika projekt.

Home	Visar vilka projekt deltagaren är ansluten till. (Windchill ProjectLink)
History	Visa upp det speciella för objektet som även inkluderar alla associerade markeringar, vad som är ändrat på objektet och all diskussion och relaterade listor till objektet.
Import	Refererar normalt till den process som skapar en ny modul i "själva databasen" genom att sända en hel katalogstruktur.
Init	Det kommandot som initierar CVS strukturen, det skapar filer och bibliotek som behövs för CVS.
Klient-server	En princip för att fördela resurser mellan datorer.
Module	En katalogstruktur. Ett mjukvaruprojekt är normalt en modul i "själva databasen".
Plan	Här visas den övergripande strukturen på projektet. Milstolparna i projektet visas här. (Windchill ProjectLink)
Release	Versionen av en fullständig produkt.
Repository	Den plats där de filer som representerar alla versioner av dina filer ligger. "själva databasen"
Revision	Versionen av en enkel fil.
SSH	Akronym för Secure SHell. Ett protokoll för säker inloggning över nätverk.
Status	I CVS visas olika färger på filerna beroende på vilken status de har, till exempel en fil är utcheckad.
Subversion	Ett grafiskt användargränssnitt till CVS, till exempel TortoiseCVS och WinCVS.
Tag	Ett symboliskt namn som ges till en uppsättning av filer vid ett specifikt tillfälle i utvecklingen.



Vad väljer din projektgrupp?
– Versionshanteringssystem för olika projekt.

Update

Tillåter dig att uppdatera information som rör ett specifikt objekt. (Windchill ProjectLink)

Hämtar alla ändringar från CVS:en gjorda i den katalog du står samt alla underliggande katalog.(CVS)

Upload file

Skapar ett dokument i en mapp. (Windchill ProjectLink)



Vad väljer din projektgrupp?
– Versionshanteringssystem för olika projekt.

Bilaga 1 Mall för momentundersökning

Windchill ProjectLink

Alla moment som snart skall utföras skall starta ifrån projektets startsida, alltså sidan "Files". Ett moment räknas när man gör ett klick.

Dessa inre funktioner ska vi använda oss av, vi ska räkna hur många moment det behövs för att utföra varje funktion.

- Upload File
- Check Out
- Check Out and Download
- Check In
- Details
 - Se vem som gjort senaste ändringen
 - Se vilket datum och vilken tid senaste ändringen är gjord
 - Se vilka kommentarer som är skrivna om senaste ändringen

CVS

Alla moment som utförs skall utgå från den plats (katalog) på den lokala datorn där filen/filerna ska användas. Ett moment räknas som ett klick.

Dessa inre funktioner ska vi använda oss av, vi ska räkna hur många moment det behövs för att utföra varje funktion.

- Add
- Check Out
- Commit
- Update
- History
 - Se vem som gjort senaste ändringen
 - Se vilket datum och vilken tid senaste ändringen är gjord
 - Se vilka kommentarer som är skrivna om senaste ändringen



Bilaga 2 Momentundersökning TortoiseCVS

Alla moment som utförs skall utgå från den plats (katalog) på den lokala datorn där filen/filerna ska användas.

Ett moment räknas som ett klick. Siffrorna inom parenteserna visar i vilken ordning varje klick görs.

Add

För att lägga till filer som vi skapat, lokalt på vår egen hårddisk, så högerklickar vi på dem (1), antingen en och en eller alla på en gång, och väljer därefter CVS Add (2). Efter det så kommer det upp en ruta där vi klickar på OK-knappen (3) och filen är tillagd.

Check out

När vi checker ut filerna laddar vi alla filer som finns i den katalogen, vi högerklickar (1) där vi vill ha filerna och väljer CVS Checkout (2), sedan klickar vi på OK-knappen (3) och den valda modulen laddas ner till katalogen.

Commit

För att uppdatera de filer som vi precis lagt till eller ändrat i, så markerar vi filen eller filerna och högerklickar (1). I menyraden som kommer upp väljer vi CVS Commit (2), ett fönster kommer upp och vi skriver in kommentarer om vilka ändringar vi gjort och klickar sedan på OK-knappen (3).

Update

Får ner alla ändringar som gjorts med filen gör vi en update. Vi högerklickar på modulen och väljer Update (1), sedan ställer vi in val på hur vi vill att den skall uppdatera och klickar på OK-knappen (2).

History

Se vem som gjort senaste ändringen

För att se vem som gjort senaste ändringen på en fil så högerklickar (1) vi på den fil vi vill ha informationen om och väljer Diff (2). Där ser vi den information som vi behöver.

Se vilket datum och vilken tid senaste ändringen är gjord

För att få fram informationen om vilket datum och vilken tid en fil senast är ändrad gör vi som ovanstående punkt, **Se vem som gjort senaste ändringen**.

Se vilka kommentarer som är skrivna om senaste ändringen

När vi vill läsa kommentarer som är skrivna om den senaste ändringen på en fil gör vi som de två ovanstående punkterna, **Se vem som gjort senaste ändringen** och **Datum och tid**. Med undantaget av att vi måste markera just den filen som vi vill se kommentarerna om (3).



Bilaga 3 Momentundersökning WinCVS

Alla moment som utförs skall utgå från den plats (katalog) på den lokala datorn där filen/filerna ska användas.

Ett moment räknas som ett klick. Siffrorna inom parenteserna visar i vilken ordning varje klick görs.

Add

En ny fil görs lokalt och skall laddas upp till servern, men den måste först addas till servern så att andra i projektgruppen kan komma åt den. Detta görs genom att man först markerar filen (1), högerklick med musen, en menyrad kommer upp och där väljer man sedan att klicka på Add (2). Efter det så klickar man på Commit (3), skriver ett log-meddelande och till sist så klickar man på en OK-knapp (4).

Check out

För att ladda ner alla filer som finns i katalogen görs en Check out. Detta görs genom att man klickar på menyn Create (1) och sedan på Checkout module (2), efter det så skriver man in modul-namnet (3) och sist så klickar man på OK-knappen (4).

Commit

Ändringar har gjorts i en fil och det är en ny version av filen, den skall checkas in på servern så att alla kan komma åt den och se ändringar som gjorts i filen. Först så markerar vi filen (1), högerklick på musen, på menyraden väljer vi att klicka på Commit (2), skriver in ett log-meddelande och klickar sedan på OK-knappen (3).

Update

Får ner alla ändringar som gjorts med filen gör vi en update. Vi högerklickar på modulen och väljer Update (1), sedan ställer vi in val på hur vi vill att den skall uppdatera och klickar på OK-knappen (2).

History

Se vem som gjort senaste ändringen

För att se vem som gjort senaste ändringen på en fil så högerklickar vi på filer och väljer Graph Selection (1). Sedan ställer vi in val och klickar på OK-knappen (2), och sist så klickar vi på den version som vi vill ha information av (3).

Se vilket datum och vilken tid senaste ändringen är gjord

För att se vilket datum och vilken tid som filen senast är uppdaterad går det till likadant som för ovanstående punkt, **Vem som gjort senaste ändringen**.

Se vilka kommentarer som är skrivna om senaste ändringen

För att se vilka kommentarer som är skrivna till den senaste versionen av en fil så gör man som de två senaste ovanstående punkterna, **Vem som gjort senaste ändringen** och **Datum och tid**.



Vad väljer din projektgrupp?
– Versionshanteringsystem för olika projekt.

Bilaga 4 Momentundersökning Windchill ProjectLink

Alla moment som du snart skall utföra skall starta ifrån projektets startsida, alltså sidan "Files".

Ett moment räknas när man gör ett klick. Siffrorna inom parenteserna visar i vilken ordning varje klick görs.

Upload File

En ny fil har gjorts lokalt på din dator och skall laddas upp till servern för Windchill ProjectLink så att alla medlemmar i projektgruppen kan komma åt den. Detta görs genom att vi klickar på en knapp som heter Upload file (1), ett nytt fönster öppnas och där vi klickar på en Bläddra-knapp (2) för att välja filen (3) som ligger lokalt på datorn trycker vi på filen och klickar Öppna (4). Efter det så klickar vi på OK-knappen (5) för att ladda upp filen i vårt projekts filbibliotek.

Check out

För att checka ut en fil som vi vill arbeta med, i detta fall finns filen redan tillgänglig på datorn, klickar vi på rullmenyn Pick an Action (1) och väljer sedan Check out (2).

Check out and Download

En fil skall checkas ut men finns inte tillgänglig på datorn. Först högerklickar vi på filen (1) väljer i menyn Save target as (2). Klickar på Spara (3) i dialogrutan som kommer upp. Efter det så gör vi en Check out som ovan. (4, 5)

Check in

När vi har arbetat färdigt med en fil, gjort ändringar i filen, så måste vi checka in den för att andra ska kunna ta del av den nya versionen av filen. Detta görs genom att klicka på rullmenyn Pick an Action (1), välja Check in (2). Ett nytt fönster öppnas och där vi klickar på en Bläddra-knapp (3) för att välja filen (4) som vi har gjort ändringar i. Klickar på Öppna (4) och sen klickar vi på OK-knappen (5) och den nya versionen av filen är nu uppdaterad i vårt projekts filbibliotek.

Details

Se vem som gjort senaste ändringen

För att se vem som gjort senaste ändringen behöver vi inte göra något efter filen står det vem som gjort senaste ändringen av filen.


Se vilket datum och vilken tid senaste ändringen är gjord

För att se vilken dag och tid ändringen på en fil senast är gjord behöver vi inte göra något, efter filen står det vilket datum och vilken tid som filen senast ändrades.

Se vilka kommentarer som är skrivna om senaste ändringen



Vad väljer din projektgrupp?
– Versionshanteringssystem för olika projekt.

För att se vilka kommentarer som är skrivna till senaste ändringen på en fil, klickar vi på  (1) som visar historiken om kommentarerna till den filen.



Vad väljer din projektgrupp?
– Versionshanteringssystem för olika projekt.

Bilaga 5 Missiv

Hej!

Vi är tre studenter på Blekinge Tekniska Högskola som i vårt examensarbete gör en undersökning om versionshantering. Denna undersökning utgör en stor del av vårt examensarbete, där vi hoppas kunna komma fram till om det har någon betydelse av valet av versionshanteringssystem.

Vi har valt att göra vår undersökning med intervjuer för att vi inte vill beskriva allt för mycket vad det är vi vill ha svar på så att Ni eventuellt skulle bli ledda av våra frågor. Därför skickar vi här med frågorna så att Ni innan har lite tid att titta igenom dem.

Vi undersöker två stycken versionshanteringssystem, Windchill Project Link och CVS. Vi skall ta reda på om vissa faktorer påverkar valet av versionshanteringssystem.

Om någon fråga skulle uppfattas som oklar, så var vänlig att hör av dig till Anneli Sandberg så snart som möjligt. E-post: dur00asa@student.bth.se,
telefonnummer: XXXX-XX XX XX

Vi vill göra er uppmärksamma på att om ni vill vara anonyma i undersökningen respekterar vi detta och kommer inte att skriva något namn.

Tack för att Ni tar Er tid att medverka i vår undersökning.!

Ulrika Karlsson
Anneli Sandberg
Sara Olofsson



Bilaga 6 Intervjufrågor

Frågor till intervjun

Projekt information

1. Vilket företag/skola tillhör Er projektgrupp?
2. Hur många personer ingår i Er projektgrupp?
3. Totalt, hur många timmar är projektet för hela projektgruppen?
4. Hur många personer i Ert projekt är det som inte har något med systemet att göra, vilka det är som inte använder systemet?

Versionshanteringssystem information

5. Vilket versionshanteringssystem använder Ni i projektgruppen?
6. Varför använder Ni just detta system?
7. Vilka funktioner använder Ni i systemet?
8. Vilka funktioner tycker ni är viktigast för det systemet Ni använder?
9. Finns det några funktioner som Ni saknar/inte använder i Ert system?



Vad väljer din projektgrupp?
– Versionshanteringssystem för olika projekt.

Bilaga 7A Intervju BTH/Karlskrona kommun

Intervju med anonym representant, BTH/Karlskrona kommun, den 27 mars 2003, klockan 11.00

1. Blekinge Tekniska Högskola i samarbete med Karlskrona Kommun
2. 20 stycken som arbetar heltid, kurs Organisationsprocesser och informationssystem.
3. 20 personer* 40 timmar* 20 veckor,
Totalt 16000 timmar.
4. 17 stycken som använder systemet och då alltså 3 som inte använder det.
5. De använder Windchill ProjectLink
6. Intervjupersonen hade tidigare använt systemet och ansåg att det var ett bra versionshanteringssystem.
7. Projektgruppen använder de funktioner som finns under Files, det vill säga Check out, Update, Check in och Details, vilket de även tycker är de mest viktiga.
De använder även: Plan, Deliverables, Forum och intervjupersonen använder även funktionerna under Home.
8. Den viktigaste funktionen enligt projektgruppen är att det är webbaserat. Vilket de även medger kan ha negativa konsekvenser som när servern ligger nere, och man kan inte komma åt sina filer.
9. De saknar inte några funktioner i systemet vad de kan komma på. De menar att detta kan beror på att de tidigare inte har använt något annat versionshanteringssystem än Windchill ProjectLink.



Vad väljer din projektgrupp?
– Versionshanteringssystem för olika projekt.

Bilaga 7B Intervju BTH/Privat företag

Intervju med Pär Svensson BTH/Privat företag den 9 april kl. 13.00

1. Stort programvaruprojekt, 20 p i samarbete med privat företag.
2. 13 stycken heltid av programvaruteknikstudenter, och 3 stycken från programmet "Människor Datateknik Arbetsliv".
3. 13 personer* 40 timmar* 20veckor =10400,
3 personer* 20 timmar* 10veckor =600,
Totalt, 11000 timmar
4. Samtliga i projektgruppen använder systemet.
5. De använder sig av CVS på servern och WinCVS på klienten. Detta använder de till all källkod och all dokumentation.
6. Det finns flera i gruppen som hade tidigare erfarenhet ifrån detta system. De ansåg sig också ha hört en del om CVS och det hade ett bra rykte.
7. De använder sig av Check out, Update, Commit, Tag och Merge.
8. De anser att ovanstående funktioner även är de viktigaste.
9. De saknar ingenting i CVS. Funktionerna som de inte använder är många eftersom CVS är så stort. Detta anser de också är en nackdel.



Vad väljer din projektgrupp?
– Versionshanteringssystem för olika projekt.

Bilaga 7C Intervju BTH/IUC – Olofström

Intervju med Carl-Johan Wåhlberg BTH/IUC - Olofström den 31 mars, klockan 13.00

1. Blekinge Tekniska Högskola Examensarbete 10 poäng.
2. De är 3 personer.
3. Kursen är på 10 poäng Maskinteknik.
3 personer*40 timmar* 10 veckor
Totalt 1200 timmar.
4. Alla använder systemet och det gör även handledaren på kursen.
5. De använder Windchill ProjectLink.
6. Kursansvarig sade att de skulle använda sig av detta verktyg, de tyckte dock att det var lite onödigt och jobbigt att behöva läras sig något nytt. De hade inte använt något versionshanteringssystem om inte kursansvarig sagt att de måste. Dock är handledaren på företaget som de jobbar mot inte intresserad av att använda Windchill ProjectLink.
7. De använder sig av funktionerna för Plan och Files. De är inne och uppdatera ca 1 gång i veckan.
8. De tycker att användningen av Plan är bäst, då de ändå sparar på disketter också.
9. Eftersom de inte sysslat med något annat liknande verktyg, så därför tycker de att de inte saknar något. De använder sig inte av Forum eller Home. De tycker det tog tid att lära sig, men det går bättre och bättre. De tycker också att systemet går för långsamt och tar mycket tid. De anser också att de har bra koll på vad som är gjort och inte gjort i projektet, via funktionen Plan.



Vad väljer din projektgrupp?
– Versionshanteringssystem för olika projekt.

Bilaga 7D Intervju BTH/Ericsson AB

Intervju med Joakim Andersson och Gabriel Falkenberg, BTH/Ericsson AB,
Ericsson AB den 31 mars 2003 klockan 14

1. Stort programvaruprojekt, 20 p, i samarbete med Ericsson AB
2. De är totalt 19 personer, varav 4 är från ekonomiprogrammet eller programmet "Människor Datateknik Arbetsliv". De fyra arbetar inte i projektet längre.
3. 15 personer* 40 timmar* 20 veckor = 12000
4personer* 20 timmar* 10 veckor = 800,
Totalt 12 800 timmar
4. De som var från ekonomiprogrammet eller programmet "Människor Datateknik Arbetsliv", 4 personer, använder inte versionshanteringssystemet. Annars använder alla andra i projektet versionshanteringssystemet.
5. De använder CVS på servern och WinCVS på klientsidan, till dokument och konfigurationsfiler. För källkoden använder de IntelliJ IDEA 3.0.
6. De har haft föreläsningar om CVS när de gjorde ett mindre projekt, det var ingen som använde sig av det då men alla visste hur det fungerade. De anser att CVS även är en standardlösning som de tycker är bra, för att det ger stöd i deras utvecklingsmiljö. CVS är även gratis och kostar inga pengar.
7. Projektgruppen säger att de använder de mest grundläggande funktionerna i CVS, Check in, Check out, Add och Commit. När de har en leverans så "taggar" de dessa filer så att de vet vilka som hör ihop med vilken leverans.
8. De anser att ovanstående funktioner är även de viktigaste.
9. Vissa saker som branch fungerar inte riktigt, det är buggar med det i systemet. Men detta är något som de skulle vilja använda sig mer av. De tycker att det är svårt att ta bort filer, programmet känns osäkert, och att användbarheten hos klienten inte är särskilt bra. De skriver inga kommentarer till filerna eller dokumenten. De tror att det säkert finns många funktioner som de inte använder sig av, men de kom inte på några. Något som de tyckte kunde vara bra att ha, är att man kan se de allra senaste versionerna på ett ställe och att dessa uppdateras i realtid.



Bilaga 7E Intervju BTH/Företag

Intervju med Julia Dogertz och Maria Lundin, BTH/Företag, Gräsvik 31 mars, 2003 klockan 10

1. BTH, de gör sitt examensjobb.
2. De är 3 stycken som skriver ihop.
3. 3 personer* 40 timmar* 10 veckor,
Totalt 1200 timmar
4. Samtliga använder sig av systemet.
5. De använder sig av Windchill ProjectLink.
6. De fick erbjudande av Anders Cronholm på CCA att använda sig av Windchill ProjectLink under sitt examensjobb, de blev inte tvingade men Anders tyckte att det kunde vara bra att ha använt sig av något versionshanteringssystem.
De tycker att det är ett bra system att använda, till exempel om det är någon som åker hem en helg har de fortfarande tillgång till filerna utan att behöva hålla på med att spara på disketter.
De har inte fått all information om systemet än utan bara en grundgenomgång och ska ha en genomgång till senare. De kände inte heller till några andra versionshanteringssystem när de började.
7. De funktioner som de använder i dagsläget är att lägga upp dokument, bilder och länkar, det vill säga funktionerna inom Files. De använder sig också av Plan med milstolpar och delmål.
8. Den viktigaste funktionen är att de kan lägga upp filer.
9. Något som de saknar är att det inte finns någon knapp där man loggar ut, man måste stänga webbläsaren för detta. Om man inte gör det och skriver in en annan url så kan man gå tillbaka och då är man fortfarande inloggad.
De funktioner som projektgruppen inte använder sig av är bland annat, Forum, Home, de har heller inte e-postat från Windchill ProjectLink
Något som kan vara dåligt med att det är webbaserat är att det ibland ligger nere. Men annars tycker de att Windchill ProjectLink verkar stabilt.



Bilaga 7F Intervju BTH/UIQ

Intervju med Roy Sandgren BTH/UIQ den 26 mars 2003, klockan 13.30

1. Stort programvaruprojekt, 20 poäng, i samarbete med UIQ.
2. 17 stycken som arbetar heltid varav 2 stycken som arbetar halvtid, från programmet "Människor Datateknik Arbetsliv".
3. 15 personer* 40 timmar* 20 veckor = 12000,
2 personer* 20 timmar* 10 veckor = 400,
Totalt 12400 timmar
4. Det är 2 stycken som har större huvudansvar för systemet, men alla är engagerade och vet hur systemet fungerar.
5. På serversida är det CVS och på klientsidan används TortoiseCVS.
På klientsidan används ett annat tredje verktyg för att få det grafiskt.
6. Det är öppet och inte knutet till någon utvecklingsmiljö, alltså det är plattformsoberoende. Roy är ansvarig för vilket versionshanteringssystem de ska använda sig av och han har arbetat med CVS innan och visste hur det fungerade. Det är lätt att hitta information om CVS och det är även lätt att få tag i hjälp om man behöver det, till exempel skriver en fråga i ett forum.
7. De funktioner som de använder är Check out, Commit, Update, Remove och Add.
8. De funktioner som de använder mest är Commit och Update.
Check out använder de när de vill testa det som finns på servern.
De tycker bra om CVS, för att flera personer kan arbeta med samma fil samtidigt. De har haft vissa missöden när en fil skall uppdateras, då systemet talat om att det blivit någon konflikt. Systemet talar då även om på vilka rader som konflikten uppstod och då får de ordna det manuellt.
9. De saknar funktionen Diff, det är att man kan se skillnad mellan de olika versionerna. Detta finns inte med från början i systemet.
En funktion som finns med i systemet men de inte använder är Taggar, man sätter taggar på filerna för att visa att de tillhör en grupp för en release.
Det finns även en funktion som heter Branch, med detta menas att man kan utveckla en version som sticker ut från en annan version. Man har en version som heter 1.3 och gör en utstickare på den versionen som då blir 1.3.1. Man kör den utstickaren som man vill testa och tittar om det blir bättre. Om det är några ändringar som man vill spara tar man med dem och gör en ny version 1.4



Bilaga 7G Intervju NN1

Intervju med anonym representant, NN1 den 1 april 2003, klockan 11.00

1. Anonym
2. De är 2 stycken programmerare, 2 till som skulle lära sig och en chef.
3. De implementerade CVS mitt under utvecklingen. De hade redan jobbat i ca 1000 timmar innan de implementerade CVS:en. Sedan jobbade de ca 1100 timmar och har 2400 timmar kvar. Totalt 4500 timmar.
4. Chefen använder det inte.
5. CVS (Concurrent Versions System) version 1.11
6. Klienten fungerar under många plattformar och finns grafisk, exempel WinCVS.
Dessutom är det gratis och som server är det lämpligt att köra något unix-liknande operativsystem, exempel FreeBSD vilket också är gratis. Dessutom är det mycket tillförlitligt och säkert, exempel via SSH kan man hemifrån komma åt CVS servern på ett säkert sätt.
7. De använder det endast för att lagra källkod/dokument och manualer.
Commit, Check out, Tag, Diff, Status, Update, Import, Admin, Add, Remove, Init
8. Säkerheten är viktig, att inga filer försvinner. Man kan lita på att när man gör en Update så skriver den inte över några filer man ändrat lokalt.
Att det går att integrera med en Bug Tracker och e-post. Exempel efter en Commit skickas en e-post till alla i projektet.
9. Det finns säkert några mer funktioner som de vill ha men inga som de kommer på, fast ibland arbetar CVS lite långsamt vilket de tycker kan vara jobbigt.
De använder inte några subversioner exempel, CVSweb, Bug Tracker.



Bilaga 7H Intervju NN2

Intervju med anonym representant, NN2 den 27 mars 2003, klockan 10.00

1. Blekinge Tekniska Högskola.
2. 6 stycken heltid, kursen Projekt som arbetsform.
3. 6 personer* 40 timmar* 10 veckor
Totalt 2400 timmar.
4. Alla använder systemet
5. De använder Windchill ProjectLink
6. Kursansvarig bad dem att använda detta system, för att de skulle få veta hur ett versionshanteringssystem fungerar.
7. De funktioner som de använder är det kan man göra under Files; Check out, Update, Check in och Details, vilka även är de viktigaste.
Använder även: Plan, Deliverables, Forum och även funktionerna under Home.
8. Viktigaste funktionen anser de är att det är webbaserat, att de kan använda och komma åt filerna överallt. Vilket projektgruppen även medger kan ha negativa konsekvenser som när servern ligger nere, och man då inte kan komma åt sina filer.
9. De saknar inga funktioner i Windchill ProjectLink, en orsak till detta är att de inte har använt något annat versionshanteringssystem.
Funktioner som inte används är Meetings, Tasks, Resources och BOM.