



Stichting NIOC

Stichting NIOC en de NIOC kennisbank

Stichting NIOC (www.nioc.nl) stelt zich conform zijn statuten tot doel: het realiseren van congressen over informatica onderwijs en voorts al hetgeen met een en ander rechtstreeks of zijdelings verband houdt of daartoe bevorderlijk kan zijn, alles in de ruimste zin des woords.

De stichting NIOC neemt de archivering van de resultaten van de congressen voor zijn rekening. De website www.nioc.nl ontsluit onder "Eerdere congressen" de gearchiveerde websites van eerdere congressen. De vele afzonderlijke congresbijdragen zijn opgenomen in een kennisbank die via dezelfde website onder "NIOC kennisbank" ontsloten wordt.

Op dit moment bevat de NIOC kennisbank alle bijdragen, incl. die van het laatste congres (NIOC2025, gehouden op donderdag 27 maart 2025 jl. en georganiseerd door Hogeschool Windesheim). Bij elkaar zo'n 1500 bijdragen!

We roepen je op, na het lezen van het document dat door jou is gedownload, de auteur(s) feedback te geven. Dit kan door je te registreren als gebruiker van de NIOC kennisbank. Na registratie krijg je bericht hoe in te loggen op de NIOC kennisbank.

Het eerstvolgende NIOC vindt plaats op 18 maart 2027 in Arnhem en wordt georganiseerd door HAN University of Applied Sciences.

Reacties over de NIOC kennisbank en de inhoud daarvan kun je richten aan de beheerder:

R. Smedinga kennisbank@nioc.nl.

Vermeld bij reacties jouw naam en telefoonnummer voor nader contact.

Grootschalig gebruik van versiebeheer in informaticaonderwijsprojecten

Jan Derriks - Domein Media, Creatie en Informatie, Hogeschool van Amsterdam

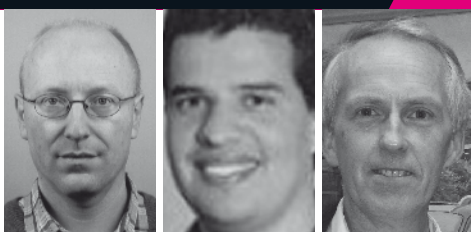
Ahmed Nait Aicha - Domein Media, Creatie en Informatie, Hogeschool van Amsterdam

Jacob Brunekreef - Domein Media, Creatie en Informatie, Hogeschool van Amsterdam

fj.derriks, a.nait.aicha, j.j.brunekreef@hva.nl

SAMENVATTING
In 2007 is bij de informaticaopleidingen van de Hogeschool van Amsterdam het gebruik van versiebeheer ingevoerd bij studentprojecten in de eerste twee studiejaar. De doelstelling was drieledig: het (leren) omgaan met een professionele werkomgeving, het bevorderen van het teamwerk in de projecten en het verhogen van het zicht van docenten op de prestaties van projectteams. We kunnen concluderen dat het eerste doel geslaagd is, het tweede doel deels geslaagd is en het laatste doel beperkt geslaagd.

TREFWOORDEN
Versiebeheer, Subversion, StatSVN, software-ontwikkeling, teamwork, teamprestaties



INLEIDING
In de IT-wereld wordt versiebeheer al langere tijd gezien als een onmisbaar instrument voor het professioneel in teamverband ontwikkelen en beheren van software. Vanaf het studiejaar 2007-2008 is aan alle teams binnen studentprojecten van de informatica-opleidingen van de Hogeschool van Amsterdam (HvA) een Subversion-repository aangeboden. De studenten werden verplicht deze versiebeheertool te gebruiken, in ieder geval bij de oplevering van de resultaten. In deze bijdrage beschrijven we onze ervaringen hiermee. We analyseren het gebruik van Subversion (svn) door teams en door individuele teamleden. De geanalyseerde data komen uit de svn-logbestanden. Voor het genereren van de svn-logbestanden hebben we gebruikgemaakt van de tool StatSVN. Met de introductie van versiebeheer hebben we drie doelstellingen voor ogen:

- 1 het leren omgaan met een professionele werkomgeving;
- 2 het ondersteunen van studenten bij (het vormgeven van) teamwork binnen onderwijsprojecten;
- 3 het verschaffen van inzicht aan docenten in het verloop van dat teamwork.

STUDENTPROJECTEN BIJ INFORMATICA

Sinds 1999 is het onderwijs bij de opleidingen Informatica en Technische Informatica van de HvA in de eerste twee studiejaar voor bijna de helft gebaseerd op grote projecten. Een project duurt een half jaar en is begroot per teamlid op circa 336 manuur (12 ECTS). Studenten werken in zelfsturende teams van circa vijf leden. Een project kent deliverables per team in de vorm van software en documenten. Een team wordt beoordeeld op deze deliverables, en op het doorlopen proces. Van ieder teamlid wordt verwacht dat hij een bijdrage levert aan de software-ontwikkeling en dat hij alle opgeleverde code kan begrijpen en toelichten.

De studentprojecten van de informaticaopleidingen kennen een aantal problemen:

- Het werk wordt vaak niet gelijk onder teamleden verdeeld. Soms kan een enkele student nagenoeg alle software hebben geschreven en zijn zijn teamgenoten niet staat om vragen over de code te beantwoorden.
- Er is vaak een te strikte scheiding in taakverdeling tussen teamleden, waarbij iedereen vooral doet waar hij al goed in is en dit niet altijd deelt met teamgenoten.
- Er wordt, ondanks een tijdig opgesteld plan van aanpak, vaak te lang gewacht met de realisatie van (deel)producten, waardoor er een piekbelasting ontstaat rond het moment van oplevering.
- De opleiding (docenten/beoordelaars) heeft onvoldoende zicht op het gevolgde software-ontwikkelproces.

Door gebruik te maken van een versiebeheersysteem onder beheer van de opleiding kan een aantal van deze problemen eerder worden gesignaleerd door zowel het team als de opleiding. Daarmee kan ook eerder aan een oplossing worden gewerkt.

VERSIEBEHEER

Sinds september 2007 wordt elke student en elk projectteam de mogelijkheid geboden om vanaf de start van het project gebruik te maken van Subversion-versiebeheer. Het onder beheer van de opleiding aanbieden van een versiebeheeromgeving biedt in onze ogen een aantal voordelen:

- 1 Veiligheid van gegevens. Een team wordt niet meer benadeeld door een teamlid dat een USB-stick verliest.
- 2 Kennisdeling wordt gefaciliteerd vanuit de opleiding en niet door bijvoorbeeld MSN.
- 3 Er wordt ervaring opgedaan met een tool die in het werkveld veel gebruikt wordt.
- 4 Versiebeheer komt de kwaliteit van de ontwikkelde software ten goede, zeker bij het werken in teamverband.
- 5 Binnen de opleidingen worden agile-ontwikkelmethoden onderwezen. Het toepassen van bijvoorbeeld test-driven development en/of daily build-and-smoke tests is nauwelijks mogelijk zonder goed ingericht versiebeheer.
- 6 Een studententeam wordt zich er meer van bewust dat het product een teamprestatie moet zijn en dat elk teamlid de code van een ander kan en mag verbeteren.
- 7 Het is mogelijk om Subversion-statistieken te verzamelen en te vergelijken voor wat betreft teams binnen één project, of over verschillende projecten. Daarmee krijgen zowel de docenten als de studenten inzage in de groei van het product en het binnen een team doorlopen ontwikkelproces.

De voordelen 1 t/m 4 hebben een algemeen inhoudelijk karakter. Ze sluiten aan bij ons eerste doel, het leren omgaan met een professionele werkomgeving. De laatste (5 t/m 7) zijn specifiek educatieve voordelen waar een opleiding gebruik van kan maken. Die dekken vooral de twee andere doelen af: studenten helpen bij het vormgeven van teamwerk en als docenten een beter zicht krijgen op dat teamwerk.

ERVARINGEN

In het studiejaar 2007-2008 zijn vanuit de opleiding 94 repositories voor teams binnen 7 projecten aangemaakt. Daarvan zijn er 12 niet gebruikt, waarvan 8 binnen één project. Hieruit concluderen we dat, uitgezonderd het ene project, er door de teams in grote meerderheid gebruikgemaakt is van de geboden faciliteit. Of er *goed* gebruik van gemaakt is, is de volgende vraag. Om een beeld te krijgen van het gebruik van Subversion door studenten is het open source tool StatSVN gebruikt. Met behulp van deze tool kunnen statistieken uit Subversion logbestanden worden gegenereerd. De statistieken hebben de vorm

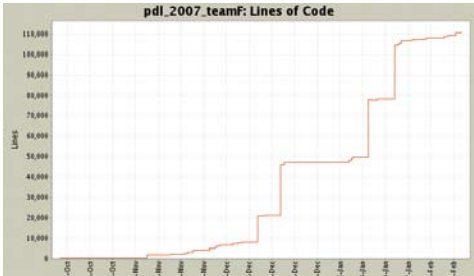


Diagram 1:
lines of code

van tabellen en grafieken. Deze statistieken kunnen gegenereerd worden per team, per project, per user en zelfs per bestand. Hieronder volgt een aantal voorbeelden van de gegenereerde statistieken.

Lines of code (LOC)

Het diagram toont het aantal ‘lines of code’ dat is geproduceerd door twee teams, gedurende de looptijd van een project. We zien dat team F (linkerdiagram) geleidelijk code heeft toegevoegd aan de repository, met enkele flinke sprongen. Team R (rechterdiagram) heeft daarentegen pas op het laatste moment (vlak voor de oplevering) de code in de repository gezet. Dit hoeft overigens niet te betekenen dat team R pas op het laatste moment is begonnen met coderen: het team kan er bewust voor gekozen hebben om pas in een zeer laat stadium gebruik te maken van Subversion. Daarmee is duidelijk dat de nodige voorzichtigheid in acht moet worden genomen bij het trekken van conclusies uit dit soort diagrammen.

Author activity

Diagram 2 geeft de verhouding weer tussen door één gebruiker nieuw toegevoegde bestanden (linkerdeel van de balk) en gewijzigde bestanden (rechterdeel van de balk). Dit diagram geeft deze verhouding weer voor vijf teamleden. We zien dat de voorlaatste gebruiker vooral wijzigingen heeft aangebracht. Dit wil niet zeggen dat hij/zij het meest heeft gewijzigd ten opzichte van andere teamgenoten; een teamlid dat slechts een keer iets wijzigt en nooit iets anders heeft gedaan zal 100% wijzigingen hebben.

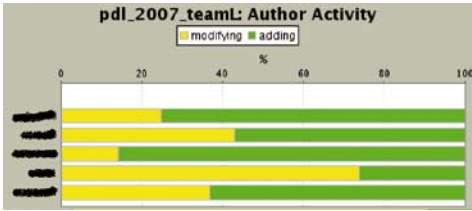


Diagram 2:
wijzigingen versus toevoegingen per teamlid

Activity by hour of day

Diagram 3 geeft de totale activiteit, uitgedrukt in het aantal commits, van het team per uur van de dag weer. We zien dat de studenten niet alleen in de geroosterde uren aan het project werken. Soms gaat het werk door tot in de kleine uurtjes.

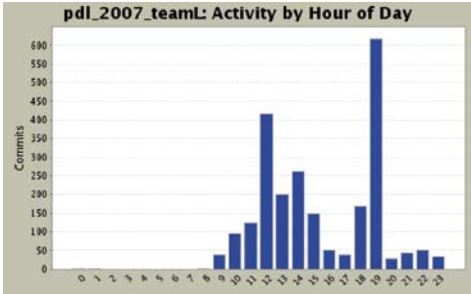


Diagram 3:
commitactiviteit per uur van de dag

Activity by day of week

In het diagram 4 zien we dat het werk aan het project niet op één dag van de week gebeurt. Sterker nog: de ingeroosterde projectdag van het betreffende project was dinsdag, terwijl de grafiek een piek op de donderdag toont.

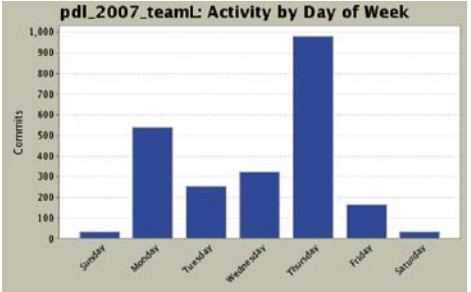


Diagram 4:
commitactiviteiten per dag van de week

ANALYSE VAN DE DATA

Om een overallbeeld te krijgen van de teamprestaties zijn deze op een aantal fronten ondergebracht in verschillende categorieën. Zo zijn de diagrammen van de groei van het aantal regels in relatie tot de tijd ondergebracht in drie categorieën: goed, matig en slecht. De onderstaande paragrafen tonen de verdeling van 82 projectteams over drie categorieën voor wat betreft verschillende aspecten van het teamwerk. Daarbij is een uitsplitsing gemaakt naar projecten uit het eerste studiejaar en projecten uit het tweede studiejaar.

Spreiding LOC over de looptijd van een project

Voor onderstaande tabel is gekeken naar de LOC-diagrammen (zie figuur 1) van alle 82 teams.

	Goed	Matig	Slecht
Teams eerste studiejaar	11	15	15
Teams tweede studiejaar	18	15	8
Totaal aantal teams	29	30	23

Wenselijk is een goede spreiding in de tijd van de werkzaamheden, zoals af te leiden uit het aantal ingecheckte regels code. De verdeling goed, matig en slecht is gebaseerd op de spreiding van activiteiten over de 21 weken van een project. De verdeling is goed als er binnen 7 weken is begonnen; slecht als er pas in de laatste 7 weken activiteit zichtbaar is en matig daartussenin.

Observaties:

- Nog niet de helft van de teams begint tijdig, dat valt niet echt mee.
- Tweedejaars doen dit beter dan eerstejaars, maar niet heel veel beter.

We concluderen dat (te) veel teams (te) laat beginnen met het opslaan van code in Subversion.

Wijzigingen ten opzichte van nieuw

Hiervoor is gebruikgemaakt van de diagrammen zoals getoond in figuur 2.

	>20%	10-20%	0-10%
Teams eerste studiejaar	21	6	14
Teams tweede studiejaar	12	12	17
Totaal aantal teams	33	18	31

Wenselijk is een significant aantal (>20%) wijzigingen t.o.v. commits van nieuwe regels code. Dat geeft aan dat een applicatie tijdens het bouwen getest en aangepast wordt.

Bij het scoren is niet alleen gekeken naar de percentage-diagrammen uit Subversion, maar ook naar de absolute aantallen activiteiten: als een teamlid 90% van alle commits voor zijn rekening neemt is het percentage in de balk van die persoon in hoge mate bepalend voor de score op dit onderdeel.

Observaties:

- Minder dan de helft van het aantal teams vertoont het gewenste gedrag, dat valt tegen.
- Een substantieel aantal blijft daar zelfs ver vandaan.
- Dit doen eerstejaars beter dan tweedejaars. Hiervoor is nog geen verklaring.

We concluderen dat (te) weinig teams Subversion gebruiken als tussenopslag voor code waaraan nog volop gewerkt wordt.

Spreiding van werkzaamheden over de week

Hiervoor is gebruikgemaakt van de diagrammen zoals getoond in figuur 4.

	Veel	Gemiddeld	Weinig
Teams eerste studiejaar	22	7	12
Teams tweede studiejaar	21	9	11
Totaal aantal teams	41	16	23

Hier is niet echt wenselijk gedrag aan te geven. De volgende verdeling is toegepast:

- Veel: minimaal drie dagen in de week met activiteit boven een denkbeeldige 30% lijn, spreiding in het uren-diagram is dan niet relevant.
- Gemiddeld: minimaal twee dagen in de week met activiteit boven een denkbeeldige 30% lijn, met duidelijke spreiding in het uren-diagram.
- Weinig of niet: overig (d.w.z. één dag in de week die eruit springt, weinig of geen spreiding in het uren-diagram).

Observaties:

- De helft van de teams werkt (redelijk) gespreid over de week.
- Er is nauwelijks verschil tussen eerstejaars- en tweedejaarsteams waarneembaar. Men zou kunnen verwachten dat tweedejaars meer spreiding vertonen omdat ze meer zelfstandiger te werk gaan en eigen tijd indelen.

Hieruit kunnen we concluderen dat veel teams min of meer zelfstandig gedrag vertonen, waarbij niet alleen gedurende de geroosterde projectdagen wordt gewerkt.

Spreiding aantal commits over teamleden

Dit onderdeel is gescoord op basis van de berekende standaard deviatie van de in Subversion getoonde percentages van het aantal commits van de afzonderlijke teamleden. Hoe kleiner de standaarddeviatie, hoe kleiner (beter) de spreiding.

- Goed: standaarddeviatie is kleiner dan of gelijk aan 15.
- Matig: de standaarddeviatie ligt tussen de 15 en 30.
- Slecht: de standaarddeviatie is groter dan 30.

	Goed	Matig	Slecht
Teams eerste studiejaar	2	9	30
Teams tweede studiejaar	6	24	11
Totaal aantal teams	8	33	41

Wenselijk is dat de spreiding over de teamleden maximaal is: iedereen doet evenveel. Dit komt overeen met de categorie ‘goed’.

Observaties:

- De helft van de teams vertoont weinig spreiding (categorie ‘slecht’): één of twee teamleden doen het werk, de rest vertoont minimale activiteiten in Subversion.
- Tweedejaars doen het gemiddeld wel duidelijk beter dan eerstejaars.
- Er zijn maar zeer weinig teams met veel spreiding van werkzaamheden, zowel in het eerste als in het tweede jaar.

Uit deze observaties kunnen we concluderen dat de spreiding van het werk (aantal commits) over teamleden erg slecht is, voor zover te zien in Subversion. Spreiding aantal regels code over teamleden Dit onderdeel is op dezelfde manier gescoord als het vorige onderdeel, met dezelfde categorieën:

	Goed	Matig	Slecht
Teams eerste studiejaar	2	9	30
Teams tweede studiejaar	3	19	19
Totaal aantal teams	5	28	49

Wenselijk is dat de spreiding over de teamleden maximaal is: iedereen doet evenveel. Dit komt overeen met de categorie ‘goed’. Voor de tweedejaars is het wenselijk dat zij het nog beter doen.

Observaties:

- De helft van de teams vertoont weinig spreiding (categorie ‘slecht’): één of twee teamleden doen het werk, de rest vertoont minimale activiteiten in Subversion.
- Tweedejaars doen het gemiddeld minder slecht dan eerstejaars.
- Er zijn maar zeer weinig teams met veel spreiding van werkzaamheden, zowel in het eerste als in het tweede jaar.

Uit deze observaties kunnen we concluderen dat de spreiding van het aantal regels code over teamleden erg slecht is, voor zover te zien in Subversion.

CONCLUSIES

Op basis van onze ervaringen met het gebruik van Subversion in studentprojecten komen we tot de volgende conclusies:

- 1
- Subversion wordt door het overgrote deel van de teams gebruikt. Van de geboden faciliteit wordt op grote schaal gebruik gemaakt. Daarbij speelt enige dwang (de verplichting om projectresultaten op te leveren via Subversion) natuurlijk wel een rol.
- 2
- Binnen projecten hebben teamleden het idee van ‘eigen code’ voor een flink deel losgelaten. Ze spreken meer over teamcode dan voorheen.

3

De intensiteit en professionaliteit van gebruik zijn zeer wisselend. We zien teams die een ‘ideaal profiel’ laten zien (regelmatige commits, goede spreiding over teamleden, collectief werken op vaste dagen/uren, goede indeling repository), maar we zien ook de nodige teams met het tegenovergestelde profiel. Hier ligt nog duidelijk een educatieve taak!

4

De statistieken die met behulp van StatSVN verzameld kunnen worden zijn niet zonder meer te gebruiken als objectief oordeel over het teamwerk.

5

Maar, de statistieken uit StatSVN hebben op zijn minst wel een grote anekdotische waarde: bij terugkoppeling van resultaten naar studenten is direct duidelijk te maken wat het verschil in werken tussen diverse teams lijkt te zijn. Voor studenten is dit zeer herkenbaar, en daarmee is het een goed startpunt voor discussies over het inrichten van projectwerkzaamheden.

Samenvattend mogen we stellen dat ons eerste doel -het kennismaken en leren omgaan met een professionele werkomgeving- geslaagd is. Studenten hebben voor het overgrote deel gebruik gemaakt van de geboden Subversionfaciliteiten. Het tweede doel -het bevorderen van het teamwerk- is deels geslaagd. Een aantal studenten geeft aan zonder meer baat gehad te hebben bij het gebruik van Subversion. Aan de andere kant suggereren bepaalde statistieken slecht teamwerk. Het derde doel -docenten inzicht verschaffen in het teamwerk- is beperkt geslaagd. Verkregen statistieken zijn niet betrouwbaar en kunnen dan ook niet zonder aanvullend onderzoek gebruikt worden bij de beoordeling van teamwerk. Aan de andere kant leveren de statistieken wel waardevol anekdotisch onderwijsmateriaal.

GERELATEERD WERK

Draheim en Pekacki ([3]) beschrijven een tool, Bloof, die op basis van gegevens uit het versiebeheersysteem CVS de evolutie van een software product analyseert. Deze tool vertoont een duidelijke gelijkenis met het ten behoeve van ons onderzoek gebruikte StatSVN. Draheim en Pekacki geven geen meetresultaten, maar geven al wel aan dat de verkregen gegevens met de nodige terughoudendheid bekeken moeten worden. Beck ([1]) motiveert het gebruik van CVS in een cursus software-engineering (SE). Doel is primair om studenten vertrouwd te maken met een professionele ontwikkelomgeving met features voor parallel ontwikkelen, gebruik van meerdere versies, goed ingericht change

management. Hij beschrijft geen ervaringen. Glassy ([4]) motiveert het gebruik van Subversion in plaats van CVS: veiligheid, robuustheid, aantal features. Hij beschrijft ook meetresultaten verkregen bij het gebruik van Subversion in SE-cursussen. StatSVN is niet gebruikt, wellicht mede hierdoor is de analyse van de meetresultaten tamelijk elementair. Burn [2] beschrijft een vergelijkbaar onderzoek als het onze: analyse van Subversion repositories in groepsprojecten op een tweetal universiteiten. Ook hij is terughoudend voor wat betreft een statistische analyse en benadrukt de anekdotische waarde van de verkregen informatie.



Frans Vodegel (links), algemeen voorzitter NIOC programmaraad met collega Jeroen Weber.



[1]

Jon Beck: Using the CVS version management system in a software engineering course. *Journal of Computing Sciences in Colleges*, vol. 20(6), pp 57-65 (2005)

[2]

Andy Burn: Analysis and Visualisation of Subversion Repositories in Group Projects. 9th Annual Conference of the Subject Centre for Information and Computer Sciences, Higher Education Academy, UK (2008)

[3]

Dirk Draheim, Lucasz Pekacki: Analytical Processing of Version Control Data: Towards a Process-Centric Viewpoint. Technical Report B-03-07, Institute of Computer Science, Free University, Berlin (2003)

[4]

Louis Glassy: Using version control to observe student software development processes. *Journal of Computing Sciences in Colleges*, vol. 21(3), pp 99-106 (2006)