

Configuration Management - Exploratie

Tim Ackermans & Dominic Voets

Opdracht 1 CM Exploratie

Configuration management (CM) houdt zich bezig met het beleid, processen en de tools voor het beheren van veranderende software systemen. Deze veranderende software systemen moeten beheert worden omdat het makkelijk is om te vergeten welke veranderingen er al zijn doorgevoerd bij elke versie. Hierbij moet ook gedacht worden aan implementatie voorstellen voor veranderingen, correctie van fouten en verandering van hardware of operating systemen. Als er geen goede CM is kunnen verschillende versies tegelijk actief in gebruik zijn of zelfs veranderd worden. CM zorgt ervoor dat er dus geen tijd verloren gaat aan aanpassingen die misschien al gedaan zijn of fouten die in andere versies al opgelost zijn. Verder kan CM voorkomen dat klanten verkeerde versies krijgen en zorgen dat altijd alle source code terug te vinden is. CM kan gebruikt worden bij individuele projecten en is vrijwel essentieel bij groepsprojecten aangezien er snel vergeten wordt welke veranderingen er zijn gemaakt. Bij CM zijn 4 hoofdactiviteiten altijd aanwezig:

1. *Change Management* Hierbij worden alle aanvragen voor verandering in de software van klanten en ontwikkelaars bij gehouden, het bepalen van kosten en invloed van het maken van deze veranderingen en de beslissing of deze veranderingen gedaan zullen worden en wanneer.
2. *Version Management* Hierbij worden er verschillende versies bijgehouden van het systeem om ervoor te zorgen dat veranderingen die gedaan zijn door verschillende ontwikkelaars niet met elkaar zullen interfereren.
3. *System Building* dit is het process van samenvoegen van alle componenten, data en libraries, en deze dan compileren en linken zodat er een uitvoerbaar systeem ontstaat.
4. *Release Management* hierbij wordt software geprepareerd voor een release naar externe klanten. Er wordt ook bijgehouden welke versie er released is voor extern gebruik.

CM voor goede software

Om goede software te maken moet deze voldoen aan een aantal eisen, hieronder beschrijven we de eisen en hoe CM hieraan kan bijdragen:

- De software moet voldoen aan de eisen van de klant.
 - Om te voldoen aan de eisen van de klant wordt het *Change Management* aspect van CM benut. Tijdens het hele traject kan rekening worden gehouden met veranderende wensen van de klant.
- De software moet de gevraagde performance hebben.
 - Aangezien de klant bepaalde resources tot zijn beschikking heeft, moet de gemaakte software ook met deze resources kunnen functioneren. Hierbij kan gedacht worden aan *System Building*. Wanneer alle componenten van een systeem samengevoegd wordt kan er relatief makkelijk gekeken worden naar performance.

- De software moet goed onderhoudbaar zijn.
 - Aan de hand van *Version Management* moet de software beheersbaar blijven. Bij goede version management zullen verschillende versies overzichtelijk blijven waardoor een andere programmeur makkelijker veranderingen aan kan brengen op de juiste plekken.
- De software moet robuust zijn voor zijn omgeving.
 - Bij robuustheid van software wordt er direct gedacht aan software die geen bugs bevat. Binnen CM zie je dit terug bij *Version Management* hierbij wordt er gekeken naar de versies die released worden. Hierbij wordt gekeken naar bugs en andere zaken die eventueel niet goed zijn.

CM en Scrum

CM en scrum gaan goed samen gezien er veel overlappende aspecten zijn. Bij scrum wordt er per sprint gekeken welk werk verricht kan worden. Ook kan een klant (product owner) zijn eisen tussen de sprints door aanpassen. Een goede Change Management moet er dan voor zorgen dat dit op een correcte manier wordt opgenomen in het project. Een ander aspect van scrum is dat na elke sprint een werkende versie wordt opgeleverd. Om dit te realiseren is een juiste Release Management nodig.

Opdracht 2 Koppeling proftaak

Verwachte problemen uit het CM Domein die verwacht worden bij de proftaak.

- Iedereen werkt tegelijk aan dezelfde code
- Samenvoegen van code
- Versiebeheer (Altijd problemen bij Git)
- Veranderende wensen van de opdrachtgever
- Bugfixes moeten bij iedereen op de PC komen

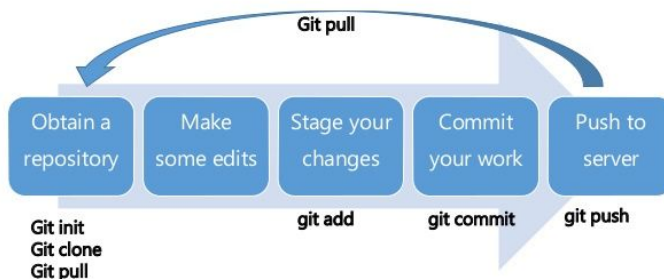
Invulling van de 3 CM activiteiten voor de proftaak

- Change management
 - Bij change management gaan we zorgen dat het probleem dat iedereen aan dezelfde code werkt oplossen. Door een goede change management bij te houden kunnen we ervoor zorgen dat radicale veranderingen geen invloed zullen hebben op andere source files. Bij dit onderdeel worden de veranderende wensen van de opdrachtgever ook meegenomen. Change management zorgt ervoor dat deze veranderingen verantwoordelijk ingebouwd kunnen worden.
- Version management
 - Bij Version management wordt het probleem van samenvoegen en versiebeheer opgelost. Doordat er regels worden gesteld aan code die gecommited mag worden zullen de merge conflicts en andere problemen voorkomen worden.
- Release management
 - Release management zal er bij ons voor gaan zorgen dat de problemen omtrent het releasen van software richting beoordelaars, klanten en andere gegadigden voorkomen zullen worden. Hierbij kan gedacht worden aan release bugs. Verder zal hier worden gecontroleerd of de goede versie is gereleased voor de correcte klant.

Inrichting & procedure voor het gebruik van source control

Voor source control gebruiken we een git. In de meest simpele variant, met alleen een master branch, ziet de flow er als volgt uit:

Git usages : Understanding Git Workflow



Om te beginnen haal je het project van server (Git pull), vervolgens kan er aan het project gewerkt worden. Deze veranderingen moeten weer teruggezet worden naar de server. Om dit te doen moeten eerst de veranderingen worden toegevoegd (git add) om deze vervolgens op te slaan (git commit). Als laatste moeten deze veranderingen naar de server gezet worden (git push). Het kan voorkomen dat er een conflict ontstaat bij het toevoegen van de nieuwe code als er een ander project lid ook code heeft toegevoegd. Deze problemen dienen zelf opgelost te worden en opnieuw via git add, git commit en git push naar de server gezet te worden.