



## Stichting NIOC en de NIOC kennisbank

Stichting NIOC ([www.nioc.nl](http://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](http://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 (NIOC2023, gehouden op donderdag 30 maart 2023 jl. en georganiseerd door NHL Stenden Hogeschool). Bij elkaar bijna 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 donderdag 27 maart 2025 in Zwolle en wordt dan georganiseerd door Hogeschool Windesheim. Kijk op [www.nioc2025.nl](http://www.nioc2025.nl) voor meer informatie.

Wil je op de hoogte blijven van de ontwikkeling rond Stichting NIOC en de NIOC kennisbank, schrijf je dan in op de nieuwsbrief via

[www.nioc.nl/nioc-kennisbank/aanmelden-nieuwsbrief](http://www.nioc.nl/nioc-kennisbank/aanmelden-nieuwsbrief)

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

R. Smedinga [kennisbank@nioc.nl](mailto:kennisbank@nioc.nl).

Vermeld bij reacties jouw naam en telefoonnummer voor nader contact.

## **Modularisering van Sterk Afhankelijke Cursussen**

### *Auteur*

Prof. Dr. Marko van Eekelen  
Radboud Universiteit Nijmegen en Open Universiteit Nederland  
Email: Marko.vanEekelen@ou.nl

### *Samenvatting*

Modularisatie van sterk afhankelijke cursussen lijkt alleen mogelijk door het onderwijs anders in te richten en de afhankelijkheden weg te nemen. Dit artikel beschrijft hoe een complex geheel van 4 software engineering cursussen (samen het 'softwarebedrijf' GipHouse van de Radboud Universiteit) gemodulariseerd kon worden. In GipHouse worden echte projecten gedaan voor echte klanten, managen studenten uit de ene cursus studenten uit de andere cursus en worden studenten geacht door te stromen van beginnend ontwikkelaar naar gevorderd manager in opvolgende cursussen. De noodzaak voor modularisatie kwam door een groeiende zij-instroom in de master en een groeiende diversificatie van het onderwijs waarbij onderdelen van het geheel van 4 cursussen op verschillende manieren ingezet kunnen worden in varianten van de opleidingen Informatica, Informatiekunde en Kunstmatige Intelligentie. Alle keuzen van de oplossing worden gemotiveerd inclusief de keuze voor RedMine boven een IBM Powerserver. Het interessante is dat op het eerste gezicht modularisatie onmogelijk is gezien de afhankelijkheid van de onderdelen van elkaar. Niettemin is het toch gelukt zonder de afhankelijkheden weg te nemen.

### *Trefwoorden*

Modularisering, Open source collaboration tools in het onderwijs, Bijzondere werkvormen, Software engineering.

# Modularisering van Sterk Afhankelijke Cursussen

*Of 'Hoe RedMine GipHouse Redde...'*

## 1 **GipHouse: een cyclus van sterk afhankelijke cursussen**

GipHouse maakt onderdeel uit van het Informatica programma van de Radboud Universiteit Nijmegen. Het is een samenhangend geheel van 4 software engineering cursussen met een bijzondere werkvorm: de studenten runnen een softwarehouse waarin ze opdrachten doen voor echte klanten. Klanten zijn andere onderwijsinstellingen maar ook IT-bedrijven (IBM, XEROX en OCE bijvoorbeeld) en natuurlijk grote en kleine software bedrijven (onder andere CapGemini en Info Support). Afhankelijk van het vak dat de student volgt, doorloopt de student verschillende rollen: van ontwikkelaar, project manager maar ook technisch manager, quality manager of directeur van het GipHouse. De leerdoelen van de cyclus zijn: samenwerking in top-down en agile projectteams, managen van top-down en agile projectteams en beheersing en toepassing van de achterliggende theoretische kernbegrippen op het gebied van de Software Engineering.

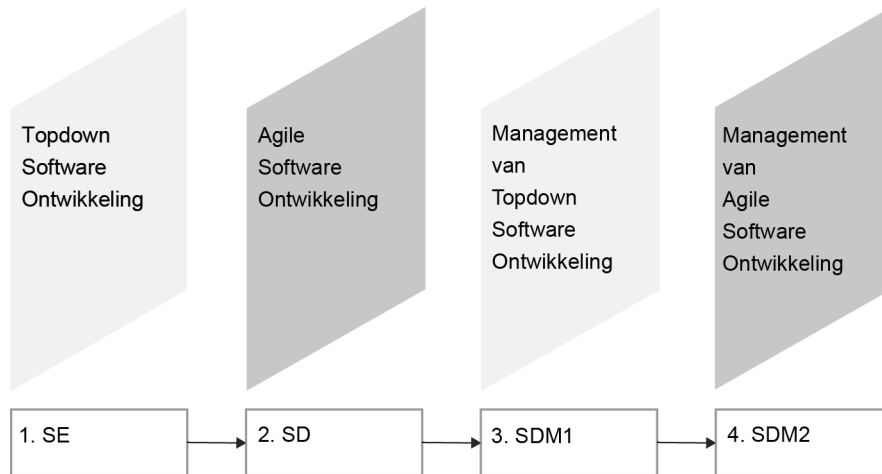
De studenten zijn in GipHouse volledig zelfstandig. De docent adviseert in principe alleen. Aangezien de docent uiteraard ook voor de beoordeling verantwoordelijk is, zijn die adviezen niet altijd vrijblijvend. De docent is echter terughoudend met het geven van strikte adviezen. Waar mogelijk (d.w.z. wanneer het niet desastreuze effecten op de eindbeoordeling zou hebben) wordt het nemen van beslissingen aan de studenten overgelaten zodat voor de studenten de volledige verantwoordelijkheid wordt ervaren voor het succesvol voltooien van de projecten.

De studenten zijn niet alleen verantwoordelijk voor het uitvoeren van de opdrachten. Ze doen ook de project werving en de projectdefinitie. De tijdens het ene semester geworven en gedefinieerde projecten worden tijdens het daaropvolgende semester uitgevoerd. Ook de indeling van de uitvoerders over de projecten en het managen van de projecten valt volledig onder de verantwoordelijkheid van de studenten. Ze organiseren extra cursussen wanneer de technische kennis voor het project ontbreekt (bijvoorbeeld over C# en .Net waar hun programmeeropleiding C++ en Java gebruikte). Ze organiseren tussentijdse voortgangbijeenkomsten en ze organiseren overleg met de klant, eindpresentaties voor de klant en overdracht van de software bij de klant.

Klanten betalen een klein bedrag voor reiskosten en andere onkosten (nadrukkelijk niet voor het opgeleverd product aangezien een succesvol resultaat in de onderwijsomgeving niet gegarandeerd kan worden). Voor de onkostenvergoeding wordt door de studenten een contract opgemaakt. De facturering geschiedt via de faculteit. De studenten zijn zelf verantwoordelijk hoe ze dit geld uitgeven. Het gaat meestal op aan reiskosten en printkosten maar ook aan koffie bij presentaties en soms een lunch bij een klantgesprek wanneer dat alleen tijdens de lunch mogelijk is. De klanten zijn over het algemeen erg tevreden over het resultaat en over de professionaliteit van de studenten.

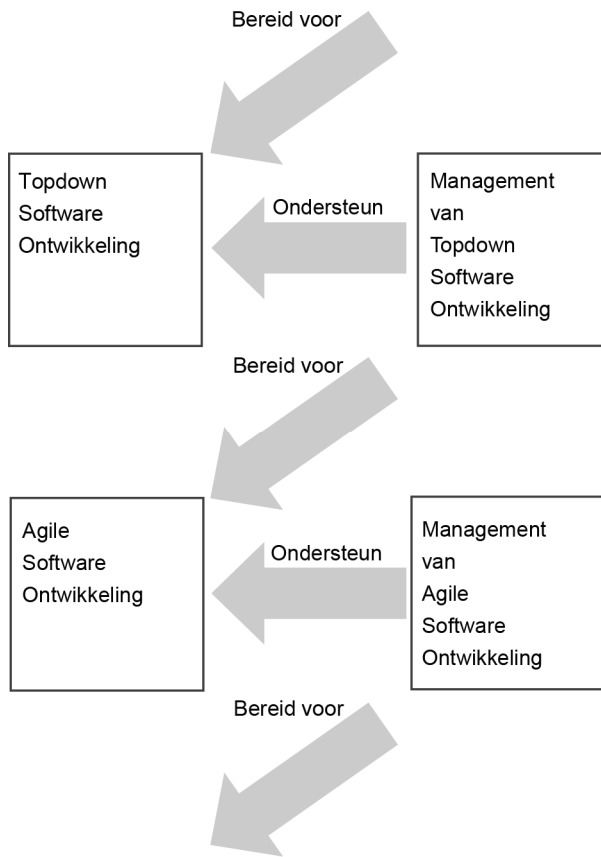
De 4 vakken waarin het GipHouse een rol speelt zijn Software Engineering (SE), Software Development (SD), Software Development Management 1 (SDM1) en Software Development Management 2 (SDM2). Het GipHouse practicum wordt in die vakken door middel van flankerend onderwijs ondersteund. Elk vak heeft naast de beoordeling van het practicum een tentamen waarin het begrip van de flankerende theorie wordt getoetst.

De studenten doorlopen de vier vakken in principe na elkaar. In Software Engineering (een derdejaars bachelor vak) zijn ze de projectuitvoerders en wordt het project volgens een top-down waterval-achtige ontwikkelmethode uitgevoerd: requirements, architectuur, functional design, technical design, implementatie. De testen worden zoveel mogelijk tijdens de eerdere fasen gedefinieerd en zo vroeg mogelijk tijdens de ontwikkeling uitgevoerd. In het daaropvolgende semester voeren ze een project uit met de, meer ervaring vereisende, agile ontwikkelmethode tijdens het Software Development vak. Vervolgens krijgen ze managementrollen in Software Development Management 1 waarin de top-down projecten van het vak Software Engineering worden gemanaged. Ten slotte, in het vierde vak van de cyclus voeren ze het management over de agile projecten in het vak Software Development Management 2.



FIGUUR 1 De vier achtereenvolgende Software Engineering onderwerpen bij de vier cursussen van GipHouse

In elk semester zijn 2 vakken tegelijk 'actief'. In het lentese semester zijn dat SE en SDM1 (top-down projecten uitvoerders en managers). In het herfstsemester zijn dat SD en SDM2 (agile). Aangezien een semester te kort is om het project te werven, te definiëren én uit te voeren, wordt in het lente semester door de studenten van SDM1 de agile projecten voorbereid en in het herfstsemester worden de top-down projecten voorbereid door de SDM2 studenten. Op deze manier is geen enkele cursus los te zien. Elke cursus is verbonden met 2 verschillende cursussen (de koppeling tussen uitvoerder en managers enerzijds en de afhankelijkheid tussen voorbereiders en uitvoerders anderzijds). Hierdoor is elk vak verbonden met 2 andere vakken: 1 in hetzelfde semester via de uitvoerders-managementkoppeling en 1 in het andere semester via de afhankelijkheid van de voorbereiding door het vak ervoor (voor de uitvoerders) of van de voorbereiding voor het vak erna (voor de managers). Dit is weergegeven in figuur 2.



FIGUUR 2 GipHouse afhankelijkheden: elk vak is gekoppeld aan 2 andere vakken

Deze cursusreeks loopt al sinds het begin van de negentiger jaren van de vorige eeuw. De studenten ervaren het als zeer positief om echte projecten te doen en ze vinden het zeer motiverend dat dit gebeurt in een omgeving waarin ze echt zelf verantwoordelijkheid dragen.

De cyclus is met recht een cyclus te noemen. Op elk punt in de reeks is een voorafgaand vak nodig om de voorbereiding ervan te doen. De koppelingen en onderlinge afhankelijkheden maken het met recht een reeks zeer sterk afhankelijke cursussen. Om die reden ligt modularisering ervan geenszins voor de hand.

## 2 Motivatie voor Modularisering

Modularisering is veelal een wens vanuit een veranderd didactisch uitgangspunt in een curriculum. In het geval van GipHouse was hiervan geen sprake. Toch hebben in de loop der jaren een aantal belangrijke wijzigingen plaatsgevonden die gezamenlijk tot de noodzaak van modularisering hebben geleid. Elk van die wijzigingen op zich rechtvaardigden niet dat modularisering nodig was maar samen bij elkaar opgeteld werd de noodzaak tot modularisering erg voelbaar.

- 1 De invoering van de Bachelor Master structuur leidde ertoe dat het SE vak een Bachelor vak werd en de andere vakken Mastervakken.
- 2 Het Master curriculum heeft vervolgens meerdere varianten gekregen waarbij slechts 1 variant het vak SD verplicht is en in alle varianten de vakken SDM1 en SDM2 keuzevakken zijn. Hierdoor zijn er studenten die een of meer van de vakken SDM1 en SDM2 willen doen, zonder dat ze SD hebben gedaan.

- 3 Alle GipHouse cursussen worden sinds de invoering van de opleiding Informatiekunde ook verplicht voor Informatiekunde studenten. Hierdoor is de achtergrond van de studenten minder uniform.
- 4 De SE cursus wordt verplicht voor Kunstmatige Intelligentie (KI)studenten. Dit maakt de achtergrond nog meer divers. Ook zijn de KI studenten van een andere faculteit en krijgen ze in de wandelgangen dus minder informatie over het bestaan en de werkwijze van GipHouse.
- 5 Vanuit het HBO is de instroom zowel in de Master Informatica als in de Master Informatiekunde aanzienlijk geworden. Die studenten hebben geen SE gedaan en ze beginnen vaak met een management vak (SDM1 of SDM2) zonder dat ze ooit een uitvoerdersvak hebben gedaan.
- 6 De laatste jaren zijn er ook steeds meer buitenlandse studenten die rechtstreeks in de Master instromen. Hun achtergrond is nog diverser dan die van het HBO. Ze hebben wel projectervaring maar in een heel andere context.
- 7 Het aantal ects punten voor het vak SE is verminderd door een curriculumaanpassing (invoering van Minoren). Dit betekent dat de voorbereiding van de projecten in SDM2 uitgebreid moet worden zodat afronding van het project binnen het vak nog mogelijk blijft.

Een van de genoemde punten op zich levert nog niet de noodzaak tot modularisatie op. Met enige improvisatie en wat kleine aanpassingen kan dat opgevangen worden.

Elk semester herrijst het GipHouse bedrijf uit zijn as. Alle werknemers zijn nieuw inclusief de directie. Dat is een bijzondere situatie. Wanneer de ruime meerderheid in de semesters daarvoor al werknemers was dan zal de 'herrijzenis' gemakkelijk plaatsvinden. Het is echter gebleken dat er grote continuïteitsproblemen op gaan treden wanneer door de vele genoemde contextwijzigingen de situatie zodanig omslaat dat het grootste deel van de werknemers en de directie bij de start voor de eerste keer van GipHouse horen. De kwaliteit van de projecten ging hierdoor achteruit. Aanpassingen kwamen niet meer aan in het volgende semester. Al met al ging de kwaliteit achteruit. Het werd nodig om het onderwijs te modulariseren zodat de 'herrijzenis' weer zonder kwaliteitsverlies kon verlopen.

Voor die modularisatie moeten wezenlijke aanpassingen in de opzet aangebracht worden zodat studenten bij elk vak kunnen beginnen en in sommige gevallen vakken ook in andere volgorde kunnen uitvoeren (dit laatste wordt niet aanbevolen en het komt weinig voor maar het wordt wel mogelijk door de modularisatie).

### 3 **Modularisatie is moeilijk!**

Het invoeren van modularisatie in GipHouse bleek een aantal bijzondere bijkomende uitdagingen op te leveren.

- Als na afloop van een semester bedacht wordt dat er volgend jaar op een bepaald punt in het proces iets verbeterd moet worden, dan moet die verbetering uitgevoerd worden door de medewerkers van een jaar later! De overdracht van die informatie moet dus 2 keer plaatsvinden: 1 keer naar het volgend semester waarin op een andere manier gewerkt wordt (topdown vs agile) en nog 1 keer naar het semester daarna waarin weer de vakken van 1 jaar terug gegeven worden. Het moet dus 2 'herrijzenissen' overleven.
- De studenten ontbeerden steeds vaker ervaring met GipHouse in eerdere cursussen. Hierdoor werd continuïteit steeds moeilijker te bereiken.
- De voertaal werd Engels. Al het materiaal moest vertaald worden wat op zich zelf weer leidde tot misverstanden door geïntroduceerde onduidelijkheden.

- De 'oude' methoden voor continuïteitsborging voldeden steeds minder. Papieren documentatie was beschikbaar maar de studenten kwamen vaak eenvoudigweg niet meer op het idee dat er ergens een papier voor zou moeten zijn. Ze gingen zelfstandig aan de slag om een oplossing te vinden maar ze liepen daardoor wel vertraging op.
- Moderne oplossingen waren nodig.

Ongeveer 5 jaar geleden is begonnen moderne oplossingen voor de continuïteit van GipHouse toe te passen.

#### 4 Wat niet werkte

Diverse oplossingen hebben de revue gepasseerd.

Websites	waren te statisch en bovendien waren ze moeilijk aan te passen. Tegen de tijd dat de studenten toegang gekregen hadden en wisten hoe aanpassingen gemaakt moesten worden was het semester bij wijze van spreken al weer half voorbij. Het opstarten hiervan was te traag.
Wiki's	snel aan te passen maar vereisen het leren van een nieuwe manier van formatteren wat weer vertragend werkt en bovendien zijn ze minder geschikt voor langere teksten.
IBM server	door IBM beschikbaar gestelde Collaboration Server met professionele software voor collaboratie. Dit was veelbelovend maar dit is na een paar jaar toch gestopt om een aantal redenen. Uitbreidingen waren beschikbaar maar het duurde steeds enkele weken voordat het geregeld was hetgeen voor de context van GipHouse een probleem vormde. Bovendien was de software (Quickr bijvoorbeeld) super voor grote organisaties maar voor GipHouse (een stuk of 50 medewerkers) was het een beetje overkill. Tenslotte hadden studenten inwerktijd nodig om er mee te leren werken hetgeen weer vertragend op de projecten werkte.

#### 5 Oplossingen die wel werken (2009-2010)

Uiteindelijk zijn in het academisch jaar 2 maatregelen genomen die samen een goede structurele oplossing vormen zodat GipHouse gemodulariseerd is en de 'herrijzenis' elk semester toch goed verloopt. Belangrijk hierbij is dat de oplossingen ook passen binnen de GipHouse context. Een docent die er extra tijd in stopt en iedereen 'bij het handje houdt', zou vast een oplossing zijn maar dat strookt niet met het GipHouse concept waarin de studenten zelfstandig het softwarehuis bestieren.

De gevonden oplossingen zijn allebei door de studenten aangedragen:

- 1 Een door de docent aangenomen en door het instituut betaalde student-assistent in de rol van *Chief Executive Officer (CEO)* die als een brug tussen de semesters functioneert en verantwoordelijk is voor de continuïteit over de semesters heen.

Om hiervoor aangenomen te worden is ervaring met tenminste 1 projectvak en 1 managementvak vereist. Deze student-assistent heeft een grote invloed in GipHouse maar het is wel een student. Hij wordt vooral aangenomen om de start van het semester goed te laten verlopen en om de zaken goed over te dragen naar het volgend semester. Tussendoor blijft hij op de achtergrond aanwezig, communicerend met de GipDirectie als 'eerste hulp' maar ook om bij te sturen waar nodig en de continuïteit in de gaten te houden onder andere door de kwaliteit van de voorbereiding voor het volgend semester te bewaken. In tegenstelling tot de studenten die precies 1 semester bij GipHouse zijn, is de student-assistent al

2 weken voorafgaand aan het semester actief in de voorbereiding van de start (in overleg met de docent) en is hij ook weer 2 weken actief in het semester daarna om te helpen bij de nieuwe start en de continuïteit te regelen (o.a. de overdracht van de projectdefinities en de geplande onderwijsverbeteringen op grond van de dit jaar geleerde lessen). Zijn taak duurt dus langer dan het semester waarin het vak loopt. Een student die het vak volgt zou alleen al daarom een dergelijke taak niet kunnen vervullen. Die wil immers aan het eind van het semester het vak afsluiten met een eindbeoordeling.

2 Een open source samenwerkingstool: *Redmine*.

Redmine levert een projectruimte (een 'gebouw') op het web waar mijlpaaldocumenten ingezet kunnen worden. Het levert ook het open source versiebeheersysteem Git, te gebruiken voor software ontwikkeling. Het bevat wiki's voor het opslaan van lesmateriaal door de docent en voor het bewaren van algemene werkinstructies en document templates. Voorts heeft het een urenregistratiesysteem via issue tracking. Hiermee wordt een volwaardige samenwerkingsomgeving gecreëerd die door de studenten goed te onderhouden is. Daarnaast is er voor extern gebruik (klanten) nog een website waarin de algemene, statische informatie over GipHouse staat.

## 6 Resultaten

De ervaringen met de CEO en het samenwerkingstool Redmine zijn uiterst positief.

- De start van projecten is niet meer vertraagd.
- De start van het semester verloopt meer gestructureerd.
- De projecten worden beter voorbereid.
- Er zijn veel minder continuïteitsproblemen.
- De projecten hebben een hogere kwaliteit
- De studenten zijn meer tevreden.
- De studenten kunnen op elk punt in de cyclus instromen en zelfs vakken in andere volgorde volgen.
- GipHouse is gemodulariseerd.

Er is echter nog een boel te verbeteren. Een dergelijk, complex samenhangend stelsel van vakken gaat pas echt goed draaien als alles op elk niveau goed in elkaar past. Dit vereist nog behoorlijk wat 'fine tuning'.

## 7 Samenvattend

Sterk afhankelijke cursussen kunnen worden gemodulariseerd door middel van toepassing van een tweetal concepten:

- 1 *Een brug* (de student-assistent) voor de continuïteit over de modules heen.
- 2 *Een gebouw* (Redmine) voor de ondersteuning van samenwerking en voor het verzorgen van de continuïteit van informatie tussen de modules.