



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 (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 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

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.

Research SW-Project

Onderzoek en softwareprojecten, water & vuur? Ervaringen met onderzoeksaspecten in softwareprojectonderwijs

Door: Marko van Eekelen, Radboud Universiteit Nijmegen / Open Universiteit.

Trefwoorden: projectonderwijs, software-ontwikkeling, onderzoek.

Het combineren van software projectwerk met onderzoek aspecten is een uitdaging. Dit artikel gaat in op ervaringen met het combineren van projectwerk met onderzoek aspecten zowel bij de Radboud Universiteit (binnen GipHouse in de Master Informatica) als bij de Open Universiteit (bij het Afstudeer Project van de Bachelor Informatica). De doelen op de twee universiteiten zijn in heel algemene zin vergelijkbaar. In concrete zin zijn er verschillen. Ook de uitwerkingen verschillen. Niet alle onderzoekleerdoelen blijken met softwareprojectwerk verenigbaar te zijn.

Inleiding

Veel Informatica opleidingen hebben een vak waarbij softwareprojectwerk centraal staat. Ook zijn er een flink aantal opleidingen waarbij aan onderzoeksaspecten aandacht wordt besteed. Als we de onderwijsleerdoelen van dergelijke vakken op een rijtje zetten, wordt duidelijk waar de knelpunten liggen.

Softwareprojectwerk heeft de volgende onderwijsleerdoelen:

- A. Projectmatig werken;
- B. Ervaring opdoen met grotere software projecten;
- C. Samenwerken.

Leerdoelen van onderzoeksaspecten zijn:

1. Literatuurstudie;
2. Onderzoeksmethoden herkennen en toepassen;
3. Onderzoeksvragen bedenken en beantwoorden.

Het inventieve van onderzoek lijkt in strijd met het productieve van het projectmatig werken aan een softwareproduct. Het is als water en vuur: onverenigbaar, tenzij je gaat photoshopen (zie figuur 1).

In dit artikel wordt ingegaan op twee experimenten om dat water en vuur toch met elkaar te verenigen.



Figuur 1. Water en vuur.

Dit betreft enerzijds het 'Afstudeerproject Bachelor Informatica' bij de Open Universiteit wat door mij samen met Frans Mofers wordt gegeven. Dit afstudeerproject beoogt beroepsgericht te zijn vanwege het karakter van de Open Universiteit (waar studenten vaak de Bachelor Informatica volgen met het doel om hun werksituatie te verbeteren), maar ook onderzoek voorbereidend omdat een Bachelor ook op een Master moet voorbereiden aangezien er ook een significant aantal studenten na het afronden van een Bachelor door gaat voor een Master. Die combinatie van doelstellingen is ingevuld door studenten een groter softwareproject te laten doen voor een klant die onderzoeker is aan de open universiteit.

Anderzijds gaat het om de cursus 'Software Development Research' die ik geef bij de Radboud Universiteit Nijmegen. SDR maakt onderdeel uit van de Master Computer Science. In SDR doen de studenten een softwareproject als bijdrage aan lopend Informatica onderzoek aan de Radboud Universiteit. Door het direct contact met de onderzoeker ontstaat een extreme vorm van agile projectontwikkelmethode.

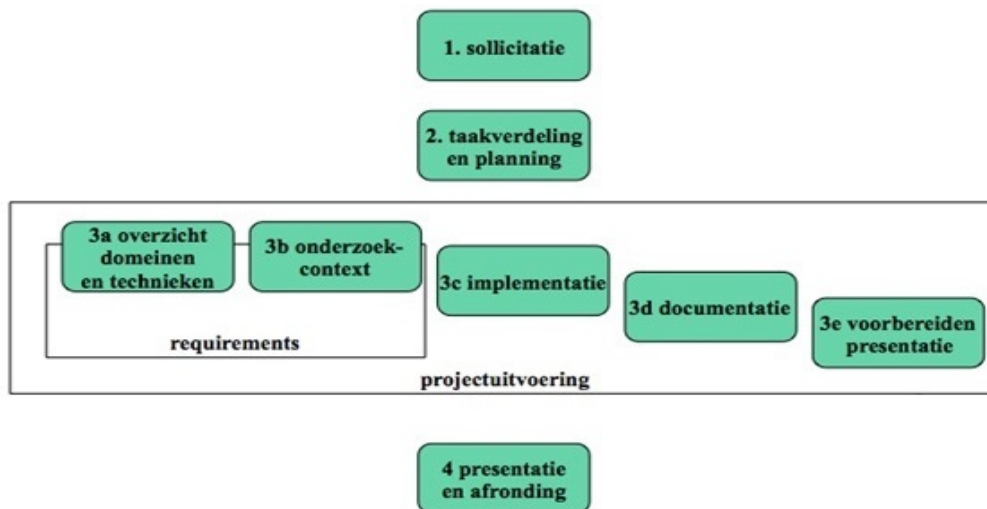
De ervaringen met beide projecten worden hieronder besproken. Risico's en te trekken lessen komen aan bod. In de conclusie zal blijken of en zo ja hoe goed water en vuur met elkaar te verenigen zijn in de verschillende contexten.

1. Afstudeerproject Bachelor Informatica (ABI) bij de Open Universiteit

Dit project vormt voor de studenten de afronding van de bachelor. Ze hebben vrijwel alle andere vakken afgerond als ze hieraan beginnen. Het is een proeve van bekwaamheid waarbij de studenten in een volledig software project, van requirements tot product, in teamverband projectmatig werkend (meestal met een agile ontwikkelmethode), hun kennis, inzicht en vaardigheden in samenhang toepassen. Het gaat niet zozeer om het verwerven van nieuwe kennis maar vooral om het integreren van de eerder opgedane kennis en het ervaring opdoen met het werken voor een echte klant. Deze klant is een onderzoeker, waardoor op natuurlijke wijze kennisgemaakt wordt met een onderzoek context. Het zelf verrichten van onderzoek is nadrukkelijk geen doelstelling van het vak.

1.1. Relevante aspecten van ABI

Effectief samenwerken is een belangrijk aspect van het projectmatig werken. Daartoe is er regelmatig teamoverleg (minimaal 1 keer per 14 dagen). Daar wordt strikt op gelet door de begeleider/docent die een coachende rol vervult. Met die coach is er ten minste een keer per maand overleg. De doorlooptijd van een project is meestal ongeveer 9 maanden. De onderzoeker is zowel klant als opdrachtgever. De studielast is 400 uur bruto per student. Het overgrote deel wordt rechtstreeks aan het project besteed. Daarnaast is er ruimte voor reflectie en onderzoekaspecten. De opbouw van de cursus is weergegeven in figuur 2.



Figuur 2. Opbouw van de cursus: Afstudeerproject Bachelor Informatica (ABI).

1.2. Recente toevoegingen aan ABI

Allereerst zijn er 2 keer per jaar mid-term-projectpresentaties ingevoerd waar bij de projecten van elkaars ervaring kunnen leren. Aangezien er zowel in september als in februari een reeks projecten start, geeft dat de gelegenheid ervaringen uit te wisselen zowel tussen projecten die in de beginfase zitten als tussen projecten die vergevorderd zijn. Daardoor ontstaat een natuurlijke feedbacksituatie. Per project worden een reeks punten besproken: waar het project over gaat, welke inhoudelijke keuzen zijn gemaakt met welke motivatie, welke uitdagingen liggen er nu en in de nabije toekomst, hoe wordt de taakverdeling geregeld, hoe is de organisatie van de samenwerking. Vanuit een kritische opstelling worden door de studenten in een constructieve sfeer vragen gesteld en wordt positieve feedback gegeven. De docenten vervullen hierbij een stimulerende en ondersteunende rol. Waar in het verleden de klanten van de projecten bedrijven of onderzoekers van buiten de faculteit waren, is er nu voor gekozen de klanten onderzoekers van de faculteit te laten zijn met de nadrukkelijke doelstelling dat de studenten kennismaken met de Informatica onderzoek context van die onderzoeker. Daartoe moeten de studenten achterhalen wat de onderzoekdoelstelling is, wat de onderzoekmethode is, hoe het softwareproject in het onderzoek past en ze moeten bedenken welke nieuwe onderzoeksvragen binnen bereik komen, dankzij het resultaat van het softwareproject. Voorbeelden van projecten zijn (bij Engelstalige onderzoekers kan de projectdoelstelling in het Engels gesteld zijn):

- Supporting hierarchical design of networks with a graphical tool;
 - Extend existing graphical support of design of multi-core chips with 50-100 cores. Current system does not scale well. Incorporate scalability features.
 - Language: C#.
 - Research context: multi-core chip design property verification for INTEL.
- Visualisatie Firewall-configuraties;
 - Doel is het realiseren van visualisaties van firewall configuraties om het onderwijs te ondersteunen. Diverse uitbreidingsmogelijkheden variërend van variaties op algoritmen, genereren van configuraties, genereren van tests en formele verificatie.
 - Language: Java.
 - Research context: improving educational quality of security teaching.

1.3. Evaluatie van ABI-onderzoekaspecten

Kort door de bocht gezegd zijn de projectleden bij de onderzoeker op bezoek. De onderzoeker is vooral klant. Er is een duidelijke rolverdeling. Het doel van het project is duidelijk. Vanuit de proeve van bekwaamheid karakter worden hoge eisen gesteld aan de documentatie. Er is overleg met de onderzoeker over vrijheidsgraden. De studenten maken kennis met het onderzoek van de onderzoeker en met de onderzoekomgeving van de onderzoeker. Hun resultaat maakt nieuw onderzoek mogelijk en ze verkennen zelf welke nieuwe mogelijkheden dit geeft. Hiermee bereiden ze zich ook voor op de keuze voor een master. Al met al is het resultaat geen water en vuur maar meer olie en azijn (zie figuur 3). Het vergt enig kloppen om het te mengen maar het brengt het gerecht op een hoger smaakniveau.



Figuur 3. Olie en azijn.

2. Software Development Research (SDR) bij de Radboud Universiteit

Het vak SDR is een Master Computer Science-vak wat in een projectvorm gegeven wordt met flankerende colleges over relevante software-ontwikkelingsaspecten zoals architectuurkeuze, formele methoden en teststrategieën. Het softwaredevelopment-project wordt met een agile-ontwikkelmethode uitgevoerd in opdracht van de onderzoeker, voor de onderzoeker, bij de onderzoeker en begeleid door de onderzoeker. De documentatie-eisen worden bepaald door wat de onderzoeker vindt dat nodig is voor het onderzoek.

2.1. Relevante aspecten van SDR

Het vak is een onderdeel van GipHouse, een al jarenlang bestaande onderwijsvorm waarbij de projecten van alle studenten samen in een 'virtueel' softwarebedrijf worden ondergebracht. Dit bedrijf wordt volledig door studenten gerund. Projecten worden bij echte klanten (o.a. IBM, OCÉ, CAP Gemini) geworven, voorbereid door ouderejaars en uitgevoerd door jongerejaars. De teams van uitvoerders worden gemanaged door ouderejaars studenten die flankerend onderwijs volgen over management van softwareprojecten.

2.2. Recente toevoegingen aan SDR

Allereerst is er in GipHouse recent een modularisering doorgevoerd [1] om de groeiende diversiteit het hoofd te bieden. Met behulp van nieuwe rollen (Chief Executive Officer die over de semesters heen actief is) en het gebruik van het open source samenwerkingstool Redmine wordt de samenwerking ondersteund tijdens de semesters en over de semesters heen wordt de continuïteit gewaarborgd.

Het vak SDR is een variant van de normale opzet waarbij ouderejaars een project in een voorafgaand semester voorbereiden met goede documentatie van architectuur, requirements en indien van toepassing een scrum backlog. Bij SDR wordt een onderzoeker gezocht die behoefte heeft aan een softwarebijdrage voor het onderzoek. Er is niet veel meer documentatie dan een half A4. De eerste stap is vaak een aantal artikelen lezen om enig idee te hebben waar het onderzoek over gaat. De studenten moet vervolgens in nauw contact met de onderzoeker de softwarebijdrage gaan invullen waarbij ze in grote mate de invulling zelf kunnen beïnvloeden. Dit kan gezien worden als een extreme variant van agile development. De klant en de uitvoerders werken letterlijk samen om het resultaat tot stand te laten komen.

Voorbeelden van projecten (voertaal van SDR is Engels):

- Security;
Developing a printing facility for personalized smart cards for the 'I Reveal My Attribute' project; requires low level device programming and understanding the research needs of the project.
- Resource analysis;
Extending the functionality of a Java resource analysis tool with functionality for creating resource analysis specific for the JDK virtual machine and adding JML support for native calls
- Formalized mathematics.
Extending the functionality of MathWiki by connecting different Math systems to each other and to the wiki.

2.3. Evaluatie van SDR-onderzoek aspecten

De studenten gaan veel verder dan op bezoek bij de onderzoeker. Ze zitten bij de onderzoeker op schoot. De onderzoeker is zowel klant als manager. De rollen lopen door elkaar en de onderzoekcontext strookt niet met een standaard projectmatige aanpak. De doelen van de software zijn vaak lastig te begrijpen, omdat de onderzoekcontext veelal complex is. De documentatie die een onderzoeker wil, is vaak minimaal ten opzichte van het te produceren product. Gebruikersinterfaces zijn vaak heel technisch, alleen gericht om functionaliteit te beïnvloeden door geavanceerde gebruikers. Ook verwacht de onderzoeker nadrukkelijk dat meegedacht wordt over hoe het onderzoek optimaal ondersteund kan worden. Je zou dit een hyper-agile-development methode kunnen noemen. In feite gaan de studenten even op in de onderzoeksgroep. Ze doen mee met het onderzoek. De resulterende software maakt nieuw onderzoek mogelijk. Het past goed in de voorbereiding van een afstudeeronderwerp. Anderzijds, wordt er binnen SDR door de studenten niet echt onderzoek gedaan. Daar is geen ruimte voor.

Gezien de sterke interactie en het ontsluiten van nieuwe onderzoeksmogelijkheden is hier zeker niet sprake van water en vuur. Maar het ontbreken van daadwerkelijke nieuwe onderzoeksresultaten binnen het project, geeft aan dat het toch een mix is met beperkingen. Thee met honing lijkt een goede karakterisering (zie figuur 4). Een paar druppels onderzoek aspecten worden toegevoegd waardoor het een stuk zoeter wordt maar het is niet een en al zoetigheid.



Figuur 4. Thee met honing.

3. Risico's, lessen en conclusies

3.1. Risico's

Het risico bestaat dat het begrijpen van de onderzoekcontext te veel tijd vraagt van de studenten om tot een resultaat te kunnen komen. Ook kan het zijn dat het de studenten niet lukt voldoende mee te denken met de onderzoeker. Niet elke student zal gemotiveerd zijn om met de onderzoeker mee te gaan. Het risico bestaat dat het resultaat toch niet van nut is voor de onderzoeker.

Bij ABI worden de risico's verkleind door een goede projectvoorbereiding in overleg met de examinerator en de onderzoeker. Bij het opstellen van het document wordt aandacht besteed aan mogelijke knelpunten die vervolgens weggenomen worden.

Bij SDR worden de risico's verkleind enerzijds doordat het een keuzevak is waardoor alleen voor onderzoek gemotiveerde studenten meedoen en anderzijds door tussentijds overleg tussen examinerator en onderzoekers. Ook worden tussentijdse presentaties gehouden in de GipHouse context waardoor de studenten feedback van elkaar krijgen.

3.2. Lessen

De belangrijkste les is uiteraard dat softwareprojecten wel degelijk ook een onderzoeksdoelstelling kunnen hebben!

In de bachelor-opleiding zijn dit:

- Kennis maken met onderzoek, met name onderzoekomgeving en onderzoekmethoden;
- Ervaren welke rol software tools spelen in onderzoek;
- Nieuwe onderzoeksvragen stellen die voortvloeien uit de nieuwe software.

In de master-opleiding zijn dit:

- Leren mee te denken met een onderzoeker zowel inhoudelijk als softwarematig;
- Leren wat anders is aan software maken in een onderzoekomgeving ('hyper'agile).

De onderzoeksdoelstelling blijft ondergeschikt aan de softwareprojectdoelstelling. De studenten leren niet onderzoekmethoden toe te passen en ze komen er niet aan toe onderzoeksvragen te beantwoorden. Die aspecten blijven gereserveerd voor cursussen zoals de afstudeeropdracht waar onderzoek de belangrijkste onderwijsdoelstelling is. Het toevoegen van onderwijsleerdoelen op het gebied van onderzoek is niettemin een duidelijke verrijking van het softwareprojectwerk: geen water en vuur maar olie met azijn of thee met suiker!

3.3. Conclusies

Het combineren van doelstellingen van onderzoek en softwareproject is grotendeels gelukt maar niet helemaal. De aspecten van onderzoeksmethoden toepassen en onderzoeksvragen beantwoorden kunnen wellicht beter aan zuivere onderzoeksprojecten overgelaten worden. In figuur 5 hieronder is weergegeven welke doelstellingen wel en welke niet gehaald kunnen worden in de combinatie van onderzoek en softwareprojectwerk.

Samen? Ja/Nee	Aspect
Ja	A. Projectmatig werken
Ja	B. Ervaring opdoen met grotere softwareprojecten
Ja	C. Samenwerken
Ja	1. Literatuurstudie
Ja	2a. Onderzoeksmethode herkennen
Nee	2b. onderzoeksmethode toepassen
Ja	3a. Onderzoeksvragen bedenken
Nee	3b. Onderzoeksvragen beantwoorden

Figuur 5. Softwareproject en Onderzoek: Wat gaat samen?

Literatuur

[1] Marko van Eekelen, 'Modularisering van sterk afhankelijke cursussen', in *NIOC 2011 - Grenzeloos ICT onderwijs. Proceedings van het 10e Nederlands Informatica Onderwijs Congres (2011)*. Stichting NIOC. ISBN 978-90-358-20340. Open Universiteit Heerlen, 2011, pp. 92-98.

Wilt u reageren op dit artikel of de presentatie? Neem dan contact op met:

Prof.dr. Marko C.J.D. van Eekelen; hoogleraar Software Technologie, Open Universiteit;

tevens verbonden aan de Radboud Universiteit Nijmegen.

marko.vaneekelen@ou.nl