



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.

From Computer Science to Software Engineering

Auteurs

Jeroen Weber
Hogeschool Utrecht
Email: jeroen.weber@hu.nl

Christian Köppe
Hogeschool Utrecht
Email: christian.koppe@hu.nl

Samenvatting

Informatica is een van de snelst veranderende disciplines. De gevolgen van de veranderingen zijn overal zichtbaar, en vragen voortdurend om het bijhouden van kennis op dit vakgebied. Uit de praktijk blijkt dat alleen kennis van de bouwstenen van Informatica, zoals programmeren en databases, niet meer voldoende is. Veel meer moeten de studenten in staat zijn om de samenhang te zien binnen de complexe processen van software ontwikkeling, daarbij wordt ook nog eens verwacht dat ze de activiteiten kunnen plaatsen in het geheel, iets dat nu nog vaak ontbreekt. Een heroriëntering van het Informatica curriculum naar een Software Engineering curriculum probeert dit gat te dichten, waarbij de internationale standaard SWEBOK een erkende basis hiervoor biedt.

In deze voordracht laten we zien hoe het curriculum van de afstudeerrichting Software Engineering aan de Hogeschool Utrecht is opgezet, en hoe deze opzet bekende problemen en tekortkomingen van klassieke Informatica opleidingen adresseert.

Kernbegrippen

Software Engineering, onderwijsvernieuwing

From Computer Science to Software Engineering

1 **Introductie**

“Hier is mijn visitekaartje. Als jullie studenten in Duitsland of Zwitserland willen komen werken, zijn ze bij ons van harte welkom”. Dit was de uitspraak van een medewerker van een van de grootste IT-bedrijven op de Duitstalige markt nadat hem was verteld hoe het nieuwe curriculum van de opleiding Informatica van de HU was opgezet. We hadden het zojuist gehad over de tekortkomingen van junior software engineers en hoe ermee moet worden omgegaan. En blijkbaar vond hij onze aanpak overtuigend.

Maar net zoals bij de conferentie in Duitsland (de SEACON in Hamburg) komen ook signalen vanuit het beroepenveld in Nederland. De beroepenveldcommissie (BVC) maakt regelmatig duidelijk waaraan de aangesloten bedrijven behoefte hebben en welke competenties beginnende IT-specialisten mee moeten brengen. De suggesties van de BVC kwamen overeen met de tekortkomingen die bij de conferentie werden besproken, het is dus een internationaal 'issue'.

De meest belangrijke tekortkomingen of suggesties voor extra aandacht in het curriculum vanuit het beroepenveld waren:

- Communicatieve vaardigheden: Schriftelijke communicatie, Presenteren en Engels
- Projectmatig werken.

De genoemde suggesties mogen niet als afzonderlijke eenheden worden beschouwd, maar moeten gerelateerd zijn aan de andere – meer technische – aandachtsgebieden. Het heeft geen zin om een projectplan op te laten stellen door software engineers om ze te leren schrijven. Dit zal de eerste tijd in de praktijk niet van ze worden gevraagd. Als er aandacht moet worden besteed aan het schrijven van een stuk tekst, kies dan bijvoorbeeld voor het schrijven van een gebruikershandleiding. Het doel blijft schriftelijk leren communiceren, maar heiligt alleen de zinvolle middelen. Studenten vinden dit erg belangrijk.

Maar niet alleen vanuit het beroepenveld – nationaal en internationaal – komen ideeën voor verbetering naar voren, ook vanuit de studenten worden verbeterpunten aangedragen. Dit wordt gedaan door standaard enquêtes, maar ook door gesprekken met kleine representatieve studentengroepen. De door hun geïdentificeerde punten waren:

- Samenhang: Ontbreken van aansluiting jaar 1 en jaar 2, projectgerelateerde vakken staan los van de technische vakken.
- Geen duidelijke profilering opleiding Informatica.
- Geen vrijheid in projecten (uitvoering).
- Verdieping: te veel verdieping in tools en niet in programmeervaardigheden.
- Verbreding: te weinig basis voor een software engineer, te beperkt in keuzevrijheden.

2 Een nieuw profiel: Software Engineer

Terwijl in sommige projecten al werd geprobeerd enkele van de gedane suggesties te realiseren, zat er in het totaal toch een structureel probleem. Een klassiek Informatica curriculum bestaat uit een aantal vakken en leerlijnen, welke voor een groot deel onafhankelijk van elkaar worden geleerd. Zo ontbreekt niet alleen de samenhang, maar is het ook erg moeilijk om bijvoorbeeld algemene beroepscompetenties goed te integreren. Er moet dus een andere, een integrale aanpak worden gekozen. Een overgang van *Informatica* naar *Software Engineering* maakt deze integratie mogelijk.

Maar wat is Software Engineering eigenlijk? Deze vraag kan op verschillende manieren worden beantwoord, maar de beste is om een internationale standaard hiervoor te gebruiken. De Software Engineering Body of Knowledge (SWEBOK) is een IEEE standaard voor het vak software engineering (Software Engineering Body of Knowledge, 2004). Deze wordt gedragen door een aantal grote internationale bedrijven zoals bijvoorbeeld: SAP, IBM, Boeing of Rational.

De SWEBOK standaard bestaat uit een aantal knowledge areas (KA), die beschrijven welke kennis bij het vak software engineering hoort. De volgende lijst geeft deze KA's weer:

- Software requirements
- Software design
- Software construction
- Software testing
- Software maintenance
- Software configuration management
- Software engineering management
- Software engineering process
- Software engineering tools and methods
- Software quality

Er is nog een aantal gerelateerde disciplines genoemd, waarvan Computer Science de belangrijkste is. Per KA worden de essentiële onderdelen beschreven, maar ook hoe deze samenhangen met andere knowledge areas. Vakinhoudelijk helpt dit bij het waarborgen van de samenhang.

SWEBOK definieert te veel aandachtsgebieden binnen het vak software engineering voor het curriculum van een vierjarige HBO opleiding. Er moet dus een keuze worden gemaakt: wat wel en wat niet. Dat lijkt een tekortkoming, maar geeft uiteraard ook mogelijkheden voor profilering van de opleiding. De keuze voor welke delen aangeboden kunnen worden hangt uiteraard ook af van de reeds aanwezige kennis. Er is daarom in eerste instantie in kaart gebracht wat er binnen het huidige curriculum aan SWEBOK elementen werd aangeboden. Opvallend hierbij was dat er met name veel aandacht was voor het aanleren van diverse technologieën, met daarentegen (te) weinig aandacht voor bijvoorbeeld software testing of software maintenance, maar ook software engineering process. Deze punten kwamen voor een groot deel overeen met de kritiek van de studenten en de opmerkingen vanuit het beroepenveld. Op basis van het overzicht was het relatief eenvoudig om inhoudelijke keuzes met betrekking tot een nieuw curriculum te maken. Op basis van de tekortkomingen, het oude curriculum en SWEBOK is een profiel opgesteld voor de Utrechts Software Engineer of Utrechtse Ingenieur met specialisatie Software Engineering: De Ingenieur die haar of zijn competenties vooral inzet voor het ontwerpen, bouwen, testen en implementeren van ict-systemen, waarbij de nadruk ligt op kwaliteit, performance en betrouwbaarheid.

Dit Utrechtse venster dat over SWEBOK is gelegd is daarna voorgelegd aan de beroepenveldcommissie, studenten en docenten en na een aantal aanpassingen beschreven in een matrix. De intensiteit van de SWEBOK aandachtsgebieden wordt met behulp van grijstinten (wit is geen intensiteit, donker is veel intensiteit) in de matrix van het Nieuwe curriculum weergegeven.

	Jaar 1 / Propedeuse				Jaar 2			
	Blok 1	Blok 2	Blok 3	Blok 4	Blok 1	Blok 2	Blok 3	Blok 4
<i>Software Requirements</i>								
<i>Software Design (SD)</i>								
<i>Software Construction</i>								
<i>Software Testing</i>								
<i>Software Maintenance</i>								
<i>Software Configuration Management</i>								
<i>Software Engineering Management</i>								
<i>Software Engineering Process</i>								
<i>Software Engineering Tools & Methods</i>								
<i>Software Quality</i>								
<i>Related Disciplines</i>								

Nieuwe curriculum, Intensiteit SWEBOK disciplines

Wat overbleef was de vraag: hoe bieden we het curriculum op een goede manier aan? Wat is een didactisch model waarin de losse leerlijnen kunnen worden vervangen door gerelateerde eenheden. En wat is een eenheid? Hoe leer je studenten te testen, of kwaliteitseisen op te stellen binnen zo'n eenheid. Wat is een goed ontwerp? Wat is goede code? Hoe maak je een project beheersbaar?

3 Een samenhangend curriculum

De samenhang in disciplines en vakgebieden die SWEBOK voorschrijft zal via het curriculum moeten worden bereikt. De Utrechtse inkleuring van SWEBOK is de Utrechtse Ingenieur met als vakgebied Software Engineering.

Voor het vaststellen van het opleidingsprofiel wordt aangesloten bij het competentiemodel zoals ontwikkeld door het HBO-I platform (Bachelor of ICT, 2009). Centraal in dit model staat de zogenaamde competentiematrix.

Voor de startende Bachelor of ICT met specialisatie Informatica en profiel Software Engineer geldt de volgende competentiematrix.

Competentiematrix Informatica Software Engineering voltijd

	<i>Analyseren</i>	<i>Adviseren</i>	<i>Ontwerpen</i>	<i>Realiseren</i>	<i>Beheren</i>
<i>Gebruikersinteractie</i>	1	1	1	1	
<i>Bedrijfsprocessen</i>	1		1	1	
<i>Software</i>	3	3	3	3	1
<i>Infrastructuur</i>		2			
<i>Hardware interfacing</i>					

Deze matrix geeft in termen van HBO-I competenties de intensiteit van de eerder getoonde matrix weer.

Een goed samenhangend curriculum vraagt om een passend didactisch model, geschikte werkvormen, en om de vorderingen zichtbaar te maken: adequate toetsing en beoordeling. In dit deel willen we kort het didactisch model schetsen, waarmee een goed samenhangend curriculum wordt neergezet. Om de schets te illustreren geven we een voorbeeld van een onderwijseenheid in het nieuwe curriculum in termen van SWEBOK en het didactische model.

Als didactisch model is gekozen voor het 4C/ID model (Coppoolse & Vroegindeweyj, 2010). De keuze hiervoor vloeit onder andere voort uit FNT beleid. Kenmerkend voor dit model is het competentie gerichte karakter van het onderwijs; studenten verwerven algemene- en beroepscompetenties door te werken aan relevante beroepsproducten. Voor het realiseren van de beroepsproducten wordt via instructional design (ID) ondersteunende informatie aangeboden, zijn er leertaken, worden deeltaakoefeningen gedaan en wordt waar nodig 'Just in time' informatie aangeboden (de 4 Componenten). De beroepsproducten worden gerealiseerd in thema's.

De thema's worden in een blok van 10 weken uitgevoerd en zijn opgebouwd uit een themaopdracht van 5 ec en twee ondersteunende cursussen van ieder 5 ec. In een themaopdracht staat altijd een beroepsproduct centraal. Dat betekent dat studenten in een themaopdracht niet alleen een technisch product opleveren, maar ook dat de studenten in een projectgroep op een professionele manier met elkaar moeten samenwerken. Samenhang wordt gewaarborgd door het feit dat het beroepsproduct centraal staat. Het product is alleen te realiseren door het volgen van de cursussen (ondersteunende informatie), waarin leertaken en oefeningen worden aangeboden. Binnen de themaopdracht kan 'Just in time' informatie worden aangeboden (bijvoorbeeld instructie over hoe schrijf je een handleiding voor het product), en is zinvolle toepassing van de algemene beroepscompetenties eenvoudig te realiseren.

De samenhang tussen de thema's wordt zichtbaar gemaakt door de technische competenties te beschrijven volgens de richtlijnen van het HBO-I. Het eindniveau kan gecontroleerd worden door eenvoudigweg de deelcompetenties van de thema's te cumuleren. De competenties zijn gerelateerd aan de SWEBOK aandachtsgebieden, waardoor samenhang in het geheel wordt bewerkstelligd. Zo komt bijvoorbeeld de HBO-I competentie Analyseren op het gebied Software in de SWEBOK disciplines Requirements Engineering, maar ook Software Design terug. Realiseren van Software (HBO-I) is vooral te vinden in Software Construction en Software Testing (SWEBOK).

4 Een voorbeeld uit het nieuwe curriculum

Het thema Software Architecture and Design wordt aangeboden in het derde blok van jaar twee van de opleiding Informatica. Het beroepsproduct is een tool waarmee business rules kunnen worden vastgelegd en getransformeerd naar verschillende platformen. De vakken Software Architecture en Pattern and Frameworks reiken daarbij als ondersteunende vakken de nodige kennis en vaardigheden aan. De belangrijkste SWEBOK disciplines in het thema zijn Software Design and Software Construction, maar er wordt ook aandacht besteed aan andere disciplines. Zo komen bijvoorbeeld de requirements weer terug, welke met behulp van de ontworpen architectuur moeten worden gerealiseerd. Ook met software engineering process wordt rekening gehouden: in de themaopdracht werken studenten volgens de Open Unified Process (OpenUP). Deze methodiek biedt vervolgens ook een goede mogelijkheid voor het integreren van de algemene beroepscompetenties (ABC), zoals beschreven in HBO-I.

Met het oog op internationalisering en in antwoord op de vraag vanuit het beroepenveld wordt het vak Patterns and Frameworks volledig in het Engels gegeven.

Hieronder volgt een overzicht van het beschreven blok.

CODE	Cursusnaam	Contact/week	Beoordeling	EC
TCIF-V2THO7-10	Themaopdracht 7 Informatica			
<i>Deel 1</i>	<i>Themaopdracht 7 Informatica</i>	4 uur theorie begeleid + 4 uur theorie onbegeleid	Assessment 80 %	4
<i>Deel 2 & 3</i>	<i>ABC & SLO</i>	Totaal ongeveer 10 contacturen	Assessment 10% & 10%	1 (0,5 & 0,5)
TCIF-V2SARCH1-10	Software Architecture	2 x (2 TH + 2PR)	Opdrachten (voldoende) + Tentamen (cijfer)	5
TCIF-V2PAFR1-10	Patterns and Frameworks	2 x (2 TH + 2PR)	Opdrachten	5

De onderwerpen zijn afgebeeld op de door het HBO-I benoemde technische competenties, zoals volgende matrix laat zien.

Niveaumatrix technische competenties op basis van HBO-I richtlijnen

	<i>Analyseren</i>	<i>Adviseren</i>	<i>Ontwerpen</i>	<i>Realiseren</i>	<i>Beheren</i>
<i>Gebruikersinteractie</i>					
<i>Bedrijfsprocessen</i>					
<i>Software</i>	2		2	2	
<i>Infrastructuur</i>					
<i>Hardware interfacing</i>					

In de studiegids is dit blok als volgt beschreven:

Het thema Software Architectuur richt zich op het ontwerpen en onderhouden van informatiesystemen. Software architectuur is een van de belangrijkste en meest uitdagende disciplines binnen Software Engineering. Software architecten vormen 'de elite' van de software engineers en ze worden veel gevraagd door het bedrijfsleven. Je leert wat zij doen, welke beslissingen zij moeten nemen en welke producten zij opleveren. Ook leer je om bij het detailontwerp en bouw binnen de kaders van een software architectuur te blijven. Daarbij worden vaak Frameworks en Design Patterns toegepast, dus is het belangrijk om daar ook voldoende van af te weten en er mee te kunnen werken.

Het beroepsproduct is een systeem waarvan de software architectuur moet worden opgeleverd en de werkende software, die conform die architectuur gebouwd is. Een aantal onderdelen in dit thema zal in het Engels plaatsvinden.

5 **Ervaringen met het nieuwe curriculum**

Tot nu toe zijn de ervaringen met het nieuwe – op SWEBOK gebaseerde – curriculum goed. Studenten zien inmiddels de samenhang en eisen deze zelfs, processen worden al wat meer als “vanzelfsprekend” gezien. Daarnaast wordt door het herhaald terug laten komen van bepaalde disciplines, zoals testen of requirements engineering, de kennis over deze onderdelen ook steviger gevestigd.

Een belangrijk punt is, dat kennis van een groot deel van de disciplines en hun onderlinge samenhang ook bij de docenten bekend moet zijn. Het is dus niet meer voldoende een aantal specialisten in een docententeam te verzamelen. Veel meer zijn echte software ingenieurs gevraagd, hier is aandacht voor nodig bij de werving en selectie, maar ook in deskundigheidsbevordering.

6 **Conclusie**

Het overstappen naar SWEBOK als basis voor een Software Engineering curriculum biedt de mogelijkheid om een aantal problemen in huidige informatica curricula te adresseren. De disciplines uit SWEBOK geven een goed referentiekader en kunnen worden gebruikt om de samenhang tussen de curriculum onderdelen te bewerkstelligen en te waarborgen.

Referenties

Bachelor of ICT, domeinbeschrijving, HBO-I stichting, 2009.

Coppoolse, R., & Vroegindewey, D. (2010). 75 Modellen van het onderwijs. Groningen: Noordhoff Uitgevers.

Software Engineering Body of Knowledge. (2004). Opgeroepen op December 17, 2010, van <http://www.computer.org/portal/web/swebok>