Van onderwijsvisie naar realisatie

Verwerkelijking binnen vakken en projecten

Inleiding

In hoofdstuk 5 van het “Opleidingsprofiel Technische Informatica” wordt een visie op onderwijs geschetst, gebaseerd op de volgende vijf uitgangspunten:

1. Studeren met plezier
2. Studeren met nieuwsgierigheid
3. Studeren binnen een contextrijke leeromgeving
4. Studeren door te leren van en met andere studenten en docenten
5. Studeren met eigenaarschap en verantwoordlijkheid

In hoofdstuk 6, 7 en 8 wordt deze visie verder uitgewerkt op niveau van leerlijnen, namelijk de praktijklijn, de kennislijn en de studentgestuurde lijn. Daarbij komt ook toetsing aan de orde.

Dit document gaat over de volgende stap, het concreet maken van de genoemde visie binnen projecten, vakken, stage en afstuderen. Om deze stap inderdaad zo concreet mogelijk te maken, wordt voorbeelden ontleend aan het huidige onderwijsaanbod. Dit onderwijsaanbod is een bewegend doel. De voorbeelden kunnen echter worden vertaald naar andere projecten en vakken.

De voorbeelden zijn vrij ver uitgewerkt, ook weer om ze zo concreet mogelijk te maken. Dat houdt niet in dat de details in steen gebeiteld staan.
De bedoeling is ideeën aan te dragen voor een haalbaar pad. Dat pad kent uiteraard vele variaties.

Voor een heldere structuur wordt als indeling van dit document de splitsing in leerlijnen aangehouden. Echter zijn ook juist de dwarsverbindingen tussen émet name de praktijklijn en de kennislijn van belang, met name voor uitgangspunt 3: Studeren binnen een contextrijke leeromgeving. De praktijklijn kan een deel van de context verschaffen die studenten nieuwsgierig maakt naar de kennislijn. Daarmee wordt ook uitgangspunt 2 geraakt.

Lift-project

Wat is het

Het lift-project is op dit moment voor alle HR-TI studenten de eerste kennismaking met een stukje praktische vakinhoud. Studenten bouwen met hun halve stamgroep een lift. (Stamgroepen, bestaande uit ca. 8 studenten, zijn een middel om studenten te helpen aansluiting vinden bij studiegenoten.) Iedere student bouwt een verdieping, bestuurd door een Arduino Uno. Het gaat daarbij om een numerieke display, knoppen en het detecteren van de liftpositie.

Daarnaast is er een centrale besturing, eveneens een Arduino, die de verdiepingen coördineert en de liftmotor aanstuurt. De verdiepings-Arduino's en de centrale Arduino communiceren met elkaar via een eenvoudig serieel protocol. Programmering gebeurt in principe in C, sommige studenten gebruiken de C++ faciliteiten die in de Arduino IDE aanwezig zijn. De verdiepingen eElk groepslid geeft minimaal 2 voorkeurstaken aan. Uiteindelijk wordt in overleg besloten. De verdeling van taken over groepsleden hoeft niet 1 op 1 te zijnn de centrale besturing moeten samenwerken, waardoor ook de studenten dit moeten doen.

De studenten gaan dit project in met een minimum aan voorkennis. De meesten kunnen net een beetje programmeren in C, netwerken en protocollen zijn nieuw voor ze en op een paar hobbyisten na weten de meesten weinig van hardware. De werkwijze is bij de meesten verkennend en ad-hoc. Goed kijken naar het logica en mechanica van een echte lift gebeurt weinig tot niet. Met een laser-cutter worden de wanden van de schacht en de liftkooi uitgesneden, waarna één en ander in elkaar wordt gezet, vaak met behulp van een lijmpistool.

Wat betreft de electronica gaan studenten aan de slag met de zogenaamde brooddoos, een bak met hardware die iedere TI student aan het begin van z'n studie krijgt. Als het geheel niet blijkt te werken wordt er veel gepiekerd en vaak lukraak geprobeerd. Soms helpen ze elkaar, soms zitten ze in hun eentje te modderen. Systematisch zoeken naar fouten, door hard- en software afzonderlijk te testen en de mogelijke foutoorzaken één voor één te elimineren is voor de meesten onbekend terrein.

Er wordt vooral veel gekeken pin-nummers en draadjes, bij dat laatste soms maar lang niet altijd geholpen door systematisch kleurgebruik. Het verschil tussen stroom en spanning, hoe je deze grootheden zou kunnen meten en waarom je dat zou doen is bij velen onbekend. Dat er ergens in het gebouw universeelmeters liggen te wachten om dichter bij de foutoorzaak te komen is bij vrijwel geen enkele student in beeld. Dat ook een ledje met een serieweerstand je hierbij zou kunnen helpen is een goed bewaard geheim.

Enerzijds zou je kunnen zeggen dat de uitgangspunten 4 en 5 hier vollop aan de orde kunnen komen. Anderzijds zitten sommigen wel erg lang in een impasse. Als je niet weet dat er systematische manieren bestaan om fouten te traceren en dat er spullen en truukjes bestaan die je daarbij kunnen

helpen, blijf je misschien worstelen zonder op het idee te komen dat andere mensen je verder zouden kunnen helpen. Je weet immers nog niet dat er een “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verder” is. Enige worsteling en spontaan ontwaken het eigen initiatief om een medestudent of docent om hulp te vragen is leerzaam. Een te lange worsteling kan zowel ontmoedigend als tijdverspillend zijn. Het is een kwestie van balans.

Veel groepen maken een soort taakverdeling, op basis van veréonderstelde aanleg. Vaak is er wel een student in het groepje die al eerder geprogrammeerd heeft. Deze neemt dan vaak de software voor z’n rekening. Soms is dat alleen de centrale software, soms programmeert diezelfde student ook alle verdiepingen. Bij vragen van een docent over de voortgang van de software wordt dan naar deze persoon verwezen, soms met enig ontzag. Ook hier zitten weer twee kanten aan. De taken verdelen is goed. Ieder op z’n eigen eiland is, blijkens uitgangspunt 4, niet de bedoeling.

Het communicatieprotocol tussen de Arduino’s is een verhaal apart. Studenten focussen vrijwel zonder uitzondering op de technische details van I^2C. De vraag *welke informatie* de verdiepingen zouden moeten uitwisselen wordt pas in laatste instantie gesteld. Bij aanvang van de studie is dit begrijpelijk. Het heeft ook te maken met de instroom. We krijgen nu eenmaal niet alleen studenten met een (in aanleg) goed abstractievermogen. Deze “details eerst” aanpak blijkt echter ook in hogere leerjaren nog door veel studenten te worden gevolgd, waardoor men zich bijvoorbeeld bij het afstuderen helemaal commit aan een bepaald merk of type hardware, die soms niet of niet snel genoeg leverbaar blijkt. Wat dat betreft zou er binnen de studie een constante aansporing moeten zijn, ontwerpproblemen wat meer vanuit een (zich ontwikkelende) helikopterview te benaderen.

En dan is er nog het aspect van de geïnvesteerde aandacht, tijd en energie. Het liftproject is niet de enige studieactiviteit. Dat een bepaalde studieactiviteit even alle aandacht opeist is een bekend fenomeen. Het antwoord van een willekeurige student op de vraag waarom zij dit of dat nog niet gedaan heeft, is vaak: eind van de week moet project zus of zo klaar zijn. Plannen blijkt iets dat geleerd moet worden. Positief punt: Al dElk groepslid geeft minimaal 2 voorkeurstaken aan. Uiteindelijk wordt in overleg besloten. De verdeling van taken over groepsleden hoeft niet 1 op 1 te zijnoende leert men vaak ook dit. Negatief punt: Indien een project of vak te veel energie wegzuigt bij andere studieactiviteiten kan uitval het gevolg zijn. Ook hier weer de noodzaak van balans.

Uiteindelijk ontstaat een lift die het wel of niet doet. Daarbinnen zijn er nuances. Liften die een enigszins efficiente volgorde van passagiers oppikken hanteren zijn in de minderheid. Liften die het helemaal niet doen gelukkig ook. Al met al leren de meesten heel wat van dit project. Een mooi project dus, met zowel technische als professional en interpersonal skills aspecten. De ideeën die nu volgen betreffende het nog meer verwerken van de besproken onderwijsvisie kunnen er een nog mooier project van maken.

Het project door de bril van de 5 uitgangspunten

1. Studeren met plezier

Als je de studenten op onze verdieping aan dit project bezig ziet, valt de informele sfeer en het contact tussen de studenten op. Het is een soort zoemende bijenkorf, zowel op het onderwijsplein als in de lokalen er omheen en ook in het stadslab. Een mooi eikpunt is de situatie tijdens de Corona lockdowns. Wat waren ze blij elkaar af en toe te zien. De meesten varen wel bij fysieke aanwezigheid en contact met studiegenoten en docenten. Voor studenten die niet te ver weg wonen is ook het dagritme een positief element: uit je bed komen en ergens heen gaan waar het gezellig is.

Daarnaast zijn velen gefascineerd door het knutselen met al die mysterieuze blokjes, busjes, draadjes en lampjes en knopjes. Ook de aansturing hiervan met software heeft voor velen z'n bekoring. Plezier in de inhoud van het vak, al is de diepte en dagelijkse werkelijkheid van dat vak nog grotendeels aan het oog onttrokken. Degenen die dat helemaal niet hebben? Misschien komt het nog, misschien past een andere studie beter bij ze. Niet alles is beïnvloedbaar.

Tenstlotte is er het plezier van de overwinning: Moeilijke problemen tegenkomen, ze één voor één oplossen en tenslotte een werkend systeem. Zelf gemaakt, samen met je team. Een vreugdevolle ervaring en positieve bekrachtiging. Yes, I can!

En er zijn ook zaken die beter kunnen. Urenlang worstelen en er niet uitkomen, waarna iemand anders het wel even in orde maakt doet afbreuk aan het plezier. Het project is voor veel eerstejaars hoog gegrepen, zelfs in z'n eenvoudigste vorm. Iets meer structuur en begeleiding zal helpen. Dat hoeft niet persee altijd 1 op 1 persoonlijke begeleiding te zijn. Ook een reader of video waarin wordt getoond hoe je zoiets nu eigenlijk aanpakt is begeleiding.

Het gaat daarbij niet om een stap voor stap "handleiding" maar meer om aanwijzingen over hoe je een zo'n combinatie-project van hard- en software behapbaar maakt. Aanwijzingen die voor docenten vanzelf spreken zoals systematisch gebruik van draadkleuren, het onafhankelijk testen van subonderdelen van zowel hardware als software en het grondig testen van de hardware voordat deze als debugging vehicle voor de software wordt gebruikt zijn zo maar wat zaken die studenten op het goede spoor kunnen zetten, zonder ze daarbij het plezier en het leereffect van zelf zoeken en vinden te ontnemen.

Wat betreft structuur kan een voorbeeld-stappenplan worden aangereikt, een beetje zoals een voorbeeld-indeling van een afstudeer- of stageverslag. Hierin worden een aantal verstandige stappen bij het maken van een dergelijke toepassing genoemd, met per stap het nut ervan en wat hints omtrent de praktische uitvoering. Een voorbeeld van zo'n stappenplan staat in onderstaande tabel.

Stap	**Waarom**	**Hoe**
Eenduidig boven tafel krijgen van de vereisten aan het projectresultaat	Helder vasleggen wat precies moet worden gemaakt	Bestuderen en samenvatten van bestaande specificatie of, indien deze niet voor handen is,

		achterhalen en vastleggen ervan
Kiezen van principe-oplossingen om het projectresultaat te verkrijgen	Zo'n principe-ontwerp maakt duidelijk welke taken er zoal zijn bij de realisatie ervan	Samen creatief nadenken en overleggen, zodat gebruik wordt gemaakt van alle ideeën binnen de groep
Inventariseren van kennis, vaardigheden en voorkeur binnen de groep	Zo kan men er voor zorgen dat ieder datgene doet waar zij goed in is of wil worden	Ieder groepslid geeft aan waar zij goed in is of wat zij wil leren.
Inventariseren van taken binnen het project	Zo kunnen deze taken worden verdeeld onder de groepsleden	Zoek taken waarvan het resultaat onafhankelijk kan worden getest
Verdelen van de taken	Doorlooptijd verkleinen	Elk groepslid geeft minimaal 2 voorkeurstaken aan. Uiteindelijk wordt in overleg besloten. De verdeling van taken over groepsleden hoeft niet 1 op 1 te zijn
Inventariseren van de benodigde hardware-onderdelen	Het kan zijn dat niet alles op voorraad is. Sommige dingen moeten zelf worden gemaakt uit basismaterialen	Opstellen van materialenlijst met bron waaruit deze kunnen worden betrokken en, indien van toepassing, kosten
Splitsen van het totale hardware systeem in onafhankelijk testbare deelsystemen	Het is dan snel duidelijk in welk deelsysteem een fout optreedt	Deelsystemen zo kiezen dat ze zo min mogelijk verweven zijn
Splitsen van de benodigde software in onderdelen	Zo kan aan die onderdelen onafhankelijk worden gewerkt en kunnen ze onafhankelijk worden getest	Maak de splitsing zo dat de onderdelen zo min mogelijk met elkaar verweven zijn
Vervaardiging en individueel testen van hardware-deelsystemen	Een systeem dat uit betrouwbare deelsystemen wordt opgebouwd is meestal zelf ook betrouwbaar	Gebruik van test-hulpmiddelen zoals een universeelmeter
Testen van de hardware als geheel	Zo verkrijgt men een stevige basis waarop in een volgende stap de software kan worden getest	Dit kan door rechtstreeks de I/O van de besturing(en) in te lezen en aan te sturen met behulp van eenvoudige testprogrammatuur
Testen van delen van de software op de grondig geteste hardware (unittest)	Zo wordt snel duidelijk waar eventuele fouten zitten	De diverse software-onderdelen aansturen via speciale, eenvoudige test-code
Testen van de samengevoegde software op de samengevoegde hardware (integratietest)	Hiermee wordt een correcte samenwerking tussen alle delen getest	Stel een testplan op aan de hand van de vereisten, voer dit stap voor stap uit en leg de resultaten vast
Opstellen gebruiksdocumentatie	Een gebruiker moet weten hoe hij het systeem kan bedienen	Bekijk het systeem vanuit het standpunt van een gebruiker

		zonder technische kennis