**Leerdoel 2:**

Schrijven is een belangrijk onderdeel van informatica, naast code schrijven moeten studenten zichzelf op allerlei manieren ontwikkelen om zich zo aantrekkelijk mogelijk te maken voor werkgevers in de IT sector. Dat alleen al is een goede reden om het te kiezen als leerdoel, maar persoonlijk had ik ook nog een andere reden: ik liep tegen mijn eigen inadequacies aan tijdens het schrijven van mijn skills documenten. Om deze redenen wilde ik graag mijn schrijfwijze verbeteren. Waarom ik nu dit schrijf.

**Methode:**

Om te zorgen dat ik de beste verbeteringstechniek te krijgen begon ik met kijken buiten mijzelf, het is immers zo dat ik niet echt een heel speciaal persoon ben. Dit zorgt voor een hele hoop mogelijkheden want ik kan veel dingen doen die anderen ook doen. Dus ben ik gaan kijken naar wat de beste manier van verbetering zou kunnen zijn. Na een tijdje nadenken bedacht ik me dat het vrij simpel was: Collectief weten mensen vaak meer dan een individu als het gaat om kritiek op 1 onderwerp en als ik wil weten wat er mis met mij is dan moet ik mezelf gaan bekritiseren. Dit is echter vrij moeilijk voor een gemiddeld mens, dus bedacht ik mij dat het veel beter is om dat aan anderen over te laten, met een soort peer review methode.

Daarom heb ik gekozen om een stuk te schrijven en vervolgens aan een aantal mensen mijn stuk te geven met een formulier en de vraag of ze comments willen toevoegen om enige andere kritiek weer te geven. Nu dat ik mijn methode van nakijken en verbetering heb, kan ik gaan kijken hoe ik dit het beste kan verkrijgen.

Het besluit was vrij snel gemaakt, argumentatie. Argumentaties zijn de moeilijkere vorm van schrijven als je het doet zonder of met zo min mogelijk bronnen. Aangezien je constant moet denken aan hoe je het moet verwoorden en hoe je het het beste kan noteren.

Vervolgens heb ik een vragenlijst opgesteld met een aantal vragen:

1. *Wat vond u van de argumenten beschreven in de Tekst? Als u een tegenargument heeft, noteer deze alstublieft ook.*
2. *Had u enige problemen met de formulering van de argumenten?*
3. *Wat vond u van de schrijfwijze (las het makkelijk/moeilijk)?*
4. *Wat vond u van de structuur van de argumentatie?*
5. *Wat vond u van de volgorde en structuur van de tekst?*
6. *Heeft u nog andere opmerkingen*

Dit zorgt ervoor dat ik een goed overzicht hebt van mijn fouten, naast de comments die achtergelaten werden door de ‘peer reviewers’ heb ik ook een goed overzicht van de fouten die heb begaan.

Als laatste moest ik nog een onderwerp kiezen. Dit is Grande Omega geworden, de reden hiervoor is een onderzoek over Grande Omega die destijds gaande was. Daarnaast was dit een ‘hot item’ en dus waren er argumenten van allerlei kanten over Grande Omega. Dit zorgt er voor dat mensen makkelijker holen kunnen zien in mijn argumentaties, wat voor meer feedback zorgt over de argumentatie.

Dan nu wat ik geschreven heb, wat volgt is het origineel en de revisie van het origineel.

**Grande Omega, de kritiek:**

Gedurende het 1e jaar informatica aan de hogeschool Rotterdam ben ik vaak tegen dingen aangelopen met betrekking tot grande omega die niet erg fijn waren. Zoals het overladen van de servers bijvoorbeeld. Dit heeft geleid tot het onderzoek dat in kwartaal 4 gehouden werd, Ditzelfde onderzoek en het feit dat ik ‘schrijven’ als een leerdoel heb gekozen heeft mij ertoe geleid om dit te schrijven, om te illustreren wat er volgens mij mis is met verschillende problemen en concepten in Grande Omega zodat ik dit kon gebruiken op 12/06/2018 voor een interview met Mevr. Anne van Brussel.

**Feedback**

Tijdens het maken van de opdrachten wordt er vaak op een specifieke manier feedback gegeven: Goed of Fout. Dit wetende, worden er complexere vraagstukken gemaakt zoals Backwards Assignments waarbij je in principe de code moet afmaken voor iemand, waarbij de werkwoorden afgekapt zijn met een . Hierbij wordt naast goed of fout; een foutmelding gegeven voor backwards assignments en wordt er aangewezen waar de uitvoering van de ingevulde code anders is dan de code van het antwoord.

Nu dat ik heb omschreven hoe de feedback werkt in de Grande Omega kan ik doorgaan met een verdere beschrijving van het probleem wat ik daarin zie, beginnende met de backwards assignments.

Interpreteren van foutmeldingen in de backwards assignments kan je een hint geven maar vaak zat weet men het probleem al maar weet men niet hoe het opgelost kan worden (dit blijkt uit de hoeveelheid vragen in school chat van IT discord), zelfs dan is de theorie op zichzelf vaak niet genoeg voor iemand om het op te lossen als men de theorie zelf niet begrijpt of zelfs verkeerd begrijpt. Het grootste probleem met grande omega is dat men niet kan achterhalen met het programma zelf wat men verkeerd doet en dus altijd naar de leraren of peercoaches (of studenten) moet om dit op te lossen. Dit is onhandig aangezien de student daarom afhankelijk is van de docent. Als metafoor zal ik een weg gebruiken die eindigt in het volledig begrijpen van de stof. Als de applicatie gelijk feedback en uitlegt aan de student wat deze verkeerd doet, dan is deze weg plat en kan je makkelijk erop lopen. Als de leraar of peercoaches constant ondervraagd moeten worden, dan is deze weg hobbelig.

Mijn kritiek over de feedback is dat het meer dynamisch kan en dat de leraar een minder prominente functie moet hebben als een student de stof wilt begrijpen en leren. Het idee dat er constant om feedback gevraagd moet worden aan docenten en peercoaches als je iets niet begrijpt lijkt mij volledig onnodig. Daarnaast, als de docent een grote rol moet spelen in het leerproces, lijkt het me beter dat de leraar de leerlingen alleen maar opdrachten geeft. Die dan in 1 of 2 lessen moeten worden gemaakt, waarbij je direct feedback krijgt. Een oplossing zou kunnen zijn dat je opgeeft en dat wordt uitgelegd per antwoord wat er verkeerd is gegaan.

**Toepassing op de hoorcolleges**

Tijdens de hoorcolleges wordt er een beeld geschetst over code met behulp van directe toepassing, vervolgens, na het hoorcollege, is het de bedoeling dat de student opdrachten gaat maken over het onderwerp besproken in het hoorcolleges. Helaas sluiten de opdrachten hier niet helemaal op aan, de besproken theorie zelf wordt niet nagekeken door het programma, bijvoorbeeld er worden geen directe vragen gesteld over Unit 4, abstracte classes en hoe je deze moet opstellen. Terwijl we dit wel in de les hebben gehad. Dit zorgt ervoor dat studenten de theorie moeilijk kunnen toepassen en dat het zeker is dat deze het begrijpen. Terwijl dit volgens mij het punt van het maken van opdrachten in Grande Omega is. Als een soort poortwachter voor de toets zodat je de theorie zeker weten begrijpt. Hoe kan je weten dat je de theorie begrijpt als grande omega niet volledig aansluit op de hoorcolleges?

Hierdoor ontstaat er een split tussen wat je moet weten en wat je krijgt te horen, de hoorcolleges leggen geen nadruk op hetgeen besproken is in grande omega, het legt specifiek theoretische concepten uit die vervolgens minimaal worden toegepast in grande omega.

**Grande Omega op een server is een slecht idee**

Grande Omega hoeft niet op een server te staan, dit zorgt voor enorme laadtijd en veel problemen, in ieder geval de toets kan eigenlijk beter op een desktop applicatie. Als voorbeeld zal ik het initiatief nemen van de Peercoach Maurice, deze heeft een alternatief gemaakt (nog in alpha) voor grande omega waarvan een aantal dingen geleerd kunnen worden, beginnende met het fixen van laadtijden en authenticaties. In zijn applicatie was het opstarten van een module, wat gelijk of zelfs moeilijker te laden is dan de backward assignments in grande omega. Dit kostte mij ~1 seconde, in vergelijking met grande omega is dit vele malen sneller. Vooral als de server veel stress krijgt, zoals u misschien al gezien heeft in de klachtenbus. Grande Omega draait niet goed als honderden mensen er gebruik van maken op exact dezelfde tijd. Sterker nog, het is soms het geval dat mensen uit de server gegooid worden tijdens het maken van de toets. Dit wordt constant onder de aandacht gebracht van de developers van Grande Omega maar er blijkt niet veel aan gedaan geweest.

Helaas brengt dit problemen met zich mee: zoals: wat als ik een gebruiker ben van een apart besturingssysteem, als deze applicatie gemaakt zou worden zou het eigenlijk voor alle besturingssystemen gemaakt moeten worden. Maar dit is technisch gezien, als concept is het superieur in vergelijking met de server.

**Grande Omega is een langzaam leermiddel**

Dit argument is vergelijkbaar met feedback maar in plaats van dat het een versterking is van het argument is het nu een argument op zichzelf.

Grande Omega is gebouwd op het maken van opdrachten, deze opdrachten dient de student met een bepaalde gedachtegang aan te pakken. Deze gedachtegang wordt niet direct geleerd aan de student, er wordt echter naar gehint door de toepassing te gebruiken tijdens de hoorcolleges. De studenten zijn immers de code van een ander aan het lezen. Structuur en volgorde worden weergegeven, wat de student dan dient over te nemen.

De methode zoals hierboven beschreven heeft alleen 1 groot probleem wat terugvalt op het argument wat ik hiervoor besproken heb over feedback. Als de student de theorie niet begrijpt of zelfs niet meer weet, is het moeilijk voor deze om op een normale manier te studeren. Als de student dan een opdracht gaat maken en vastloopt, moet deze hulp vragen om de opdracht af te kunnen maken. Daarnaast kan je niet zomaar even het goede antwoord opzoeken. Hierdoor loopt de student nog meer vast. Zodra de student iets fout heeft gedaan duurt het ontzettend lang voordat deze kan begrijpen wat er verkeerd is gegaan. Dit zorgt ervoor dat alleen de studenten die alles begrijpen kunnen leren van Grande Omega, terwijl Grande Omega juist zou moeten zorgen dat de student het begrijpt als een soort poortwachter.

Gegeven dat degene die het wel begrijpen makkelijker opdrachten kunnen maken, betekent het dus dat de studenten die de meeste profijt van grande omega hebben degene zijn die de theorie al wel begrijpen. Dit is echter achterstevoren, als de studenten die het begrijpen de opdrachten al kunnen maken dan zijn deze vrij snel klaar met het blok. Alleen zijn de studenten die het niet begrijpen en dus zeker de extra hulp nodig hebben volledig achterwege gelaten. Deze kunnen immers niet de opdrachten maken als zij de theorie niet begrijpen omdat Grande Omega deze niet verteld wat zij verkeerd hebben gedaan en waarom. Studenten kunnen op deze manier alleen leren door constant anderen op te zoeken die het wel weten of met behulp van redenatie als ze met Trail en Error het antwoord vinden(wat allerlei risico’s oplevert).

**Slot**

Kortom, Grande Omega gebruikt een aantal concepten die niet goed werken op de student, waaronder de kleine hoeveelheid feedback die de student krijgt als deze iets fout doet en de aparte manier van de toepassing van de hoorcolleges op Grande Omega. Daarnaast is het leermiddel niet iets gemaakt voor de student die misschien enige hulp nodig heeft en is het ronduit langzaam als een opdracht niet goed is en dan als kers op de taart, het werkt op een server die vaak genoeg tijdens toet situaties vastloopt.

**Revisie**

**Grande Omega, de kritiek:**

Gedurende het 1e jaar informatica aan de hogeschool Rotterdam ben ik vaak tegen problemen aangelopen met betrekking tot grande omega die slecht uitgevoerd waren. Zoals het overladen van de servers bijvoorbeeld. Na een aantal maanden kwam er bericht dat er een onderzoek gehouden zou worden naar de effectiviteit van grande omega en het gebruik daarvan in het vak project waarbij er de bedoeling is dat de student een app maakt. Ditzelfde onderzoek en het feit dat ik ‘schrijven’ als een leerdoel heb gekozen is de reden waarom ik dit stuk nu schrijf. Om te illustreren wat er volgens mij mis is met de uitvoering van Grande Omega zodat ik dit kon gebruiken op 12/06/2018 voor een interview met Mevr. Anne van Brussel wat onderdeel was van het onderzoek waar ik het zojuist over hebt gehad..

**Feedback**

**Hoe het werkt**

Tijdens het maken van opdrachten in Grande Omega wordt er door het opdrachtensysteem vaak op een specifieke manier feedback gegeven: Goed of Fout. Dit wetende, worden er complexere vraagstukken gemaakt zoals Backwards Assignments waarbij je in principe de code moet afmaken voor iemand, waarbij de werkwoorden afgekapt zijn met een . Hierbij wordt naast goed of fout een foutmelding gegeven voor backwards assignments en wordt er aangewezen waar de uitvoering van de ingevulde code anders is dan de code van het antwoord.

**Problemen**

Nu dat ik heb omschreven hoe de feedback werkt in de Grande Omega kan ik doorgaan met een verdere beschrijving van het probleem wat ik daarin zie, beginnende met de backwards assignments.

Foutmeldingen in de backwards assignments worden gegeven zodat de student kan identificeren wat het probleem is. Daarvoor moet de student wel het stuk tekst, wat de foutcode voor moet stellen, interpreteren. Deze foutcodes hebben alleen een vrij groot probleem, ze geven ten eerste vaak dingen aan die niets met de geschreven code te maken heeft. Bijvoorbeeld je krijgt de error: “you have not declared string[]” terwijl string een functie is binnen de programmeertaal die je schrijft, vervolgens kan je niet kijken waar dit fout is gegaan omdat dit fout gaat bij de typechecker en dus kan je je code niet fixen en moet je gaan puzzelen om te kijken wat er mis is. Het tweede probleem wat hier speelt is dat de student het probleem al weet maar niet hoe hij deze moet oplossen, zelfs al leest deze heel de theorie opnieuw. Zeker tijdens oefenen is dit onhandig.

Het grootste probleem met grande omega is dat men niet kan achterhalen met het programma zelf wat men verkeerd doet en dus altijd naar de leraren of peercoaches (of studenten) moet om dit op te lossen. Dit is onhandig aangezien de student daarom afhankelijk is van de docent. Als metafoor zal ik een weg gebruiken die eindigt in het volledig begrijpen van de stof. Als de applicatie gelijk feedback en uitlegt aan de student geeft om weer te geven wat deze verkeerd doet, dan is deze weg plat en kan je makkelijk erop lopen. Als de leraar of peercoaches constant ondervraagd moeten worden, dan is deze weg hobbelig.

Mijn aanbeveling voor de feedback is dat de leraar een minder prominente rol moet hebben in het leerproces van de student. Het idee dat er constant om feedback gevraagd moet worden aan docenten en peercoaches als je iets niet begrijpt lijkt mij volledig onnodig. Daarnaast, als de docent een grote rol moet spelen in het leerproces, lijkt het me beter dat de leraar de leerlingen alleen maar opdrachten geeft. Die dan in 1 of 2 lessen moeten worden gemaakt, waarbij je direct feedback krijgt. Op die manier krijg je hoe dan ook uitleg over hoe de stof werkt of je het goed hebt gedaan of niet.

**Toepassing op de hoorcolleges**

Tijdens de hoorcolleges wordt er een beeld geschetst over code met behulp van directe toepassing. Vervolgens, na het hoorcollege, is het de bedoeling dat de student opdrachten gaat maken over het onderwerp besproken in het hoorcolleges. Helaas sluiten de opdrachten hier niet helemaal op aan. Sterker nog, de besproken theorie zelf wordt niet nagekeken door het programma, bijvoorbeeld er worden geen directe vragen gesteld over Unit 4, abstracte classes en hoe je deze moet opstellen. Wat leidt tot vastlopen van de studenten als deze de theorie niet begrijpen. Daarnaast is het dan ook niet duidelijk wat er mis is gegaan. Terwijl we dit wel in de les hebben gehad. Dit zorgt ervoor dat studenten de theorie moeilijk kunnen toepassen en dat het zeker is dat deze het begrijpen. Terwijl dit volgens mij het punt van het maken van opdrachten in Grande Omega is. Als een soort poortwachter voor de toets zodat je de theorie volledig begrijpt. Hoe kan je weten dat je de theorie begrijpt als grande omega niet volledig aansluit op de hoorcolleges?

Hierdoor ontstaat er een kloof tussen wat je moet weten en wat je krijgt te horen, de hoorcolleges leggen geen nadruk op hetgeen besproken is in grande omega. Daarnaast worden er in hoorcolleges specifiek theoretische concepten uitgelegd die vervolgens minimaal worden toegepast in Grande Omega.

**Grande Omega is een langzaam leermiddel**

Dit argument sluit aan op hetgeen gezegd in het eerste stuk over feedback.

Grande Omega is gebouwd met als doel het maken van opdrachten hierin wordt de methode van schrijven constant herhaald, er zit altijd dezelfde structuur in: globale variabelen, class functies en code die iets doet met hetgeen erboven. De studenten zijn dus de code van een ander aan het lezen. Deze structuur moet de student dus overnemen.

De methode zoals hierboven beschreven heeft alleen 1 groot probleem wat terugvalt op het argument wat ik hiervoor besproken heb over feedback. Als de student de theorie niet begrijpt of zelfs niet meer weet, is het moeilijk voor deze om op een normale manier te studeren. Als de student dan een opdracht gaat maken en vastloopt, moet deze hulp vragen om de opdracht af te kunnen maken. Daarnaast kan je niet zomaar even het goede antwoord opzoeken. Hierdoor loopt de student nog meer vast. Zodra de student iets fout heeft gedaan duurt het ontzettend lang voordat deze kan begrijpen wat er verkeerd is gegaan. Dit zorgt ervoor dat alleen de studenten die alles begrijpen kunnen leren van Grande Omega, terwijl Grande Omega juist zou moeten zorgen dat de student het begrijpt als een soort poortwachter.

Gegeven dat degene die het wel begrijpen makkelijker opdrachten kunnen maken, betekent het dus dat de studenten die de meeste profijt van grande omega hebben degene zijn die de theorie al wel begrijpen. Dit is echter achterstevoren, als de studenten die het begrijpen de opdrachten al kunnen maken dan zijn deze vrij snel klaar met het blok. Alleen zijn de studenten die het niet begrijpen en dus zeker de extra hulp nodig hebben volledig achterwege gelaten. Deze kunnen immers niet de opdrachten maken als zij de theorie niet begrijpen omdat Grande Omega deze niet verteld wat zij verkeerd hebben gedaan en waarom. Studenten kunnen op deze manier alleen leren door constant anderen op te zoeken die het wel weten of met behulp van redenatie als ze met Trial en Error het antwoord vinden(wat allerlei risico’s oplevert).

**Slot**

Kortom, Grande Omega gebruikt een aantal concepten die niet goed werken voor de student, waaronder de kleine hoeveelheid feedback die de student krijgt als deze iets fout doet en de aparte manier van de toepassing van de hoorcolleges op Grande Omega. Daarnaast is het leermiddel niet gemaakt voor de student die misschien enige hulp nodig heeft en is het ronduit langzaam als een opdracht niet goed is en dan als kers op de taart, het werkt op een server die vaak genoeg tijdens toets situaties vastloopt.

Uiteindelijk ligt het allemaal aan de student, verandering en verbetering kan alleen gemaakt worden met behulp van feedback en de student is de enige die dit kan geven als je Grande Omega ziet als leermiddel.

Resultaten:

Laat ik beginnen met het grote resultaat, ik vind dat ik h. De reden hiervoor is omdat ik een grote hoeveelheid feedback heb gekregen wat mij een hoop inzicht heeft gegeven in mijn zwakke -en sterke kanten in schrijven. Om deze echt heel goed te kunnen halen moet ik veranderingen gaan maken in meer dan alleen mijn schrijfwijze maar ook in mijn persoonlijkheid aangezien ik aardig slordig ben.

Kortom heb ik problemen met de volgende dingen:

* Structureel mijn stuk opstellen.
* Mijn argument logisch opstellen
* Leerbaarheid was minder, dit houdt in dat ik soms punten vergat en woorden of dat mijn zinnen te lang zijn.
* Enige problemen met witte stukken in mijn tekst

Daarnaast heb ik ook nog een sterk punt:

* Inhoudelijk waren de argumenten goed.

**Reflectie:**

Dit onderwerp had misschien beter gegaan als ik buiten de schrijfruimte had gestapt en mijzelf op een andere manier probeer te veranderen. Ook had het beter gegaan als ik meer geschreven had, op deze manier had ik misschien meer feedback kunnen krijgen.