

## Procesverslag van Ronald de Haan

Stagiair van de Hogeschool Rotterdam



|                         |                                     |
|-------------------------|-------------------------------------|
| Student                 | Ronald de Haan                      |
| Studentnummer           | 0809766                             |
| Opleiding               | Informatica                         |
| School                  | Hogeschool Rotterdam                |
| Stagebedrijf            | Quadrant Software bv                |
| Stagebegeleider bedrijf | Martin Bos (martin.bos@quadrant.nl) |
| Stagebegeleider school  | Bob Joziase (b.joziase@hro.nl)      |
| Datum                   | 24 september 2011                   |
| Documentstatus          | Definitief                          |



## Inhoudsopgave

|   |    |
|---|----|
| 1. Inleiding.....   | 3  |
| 2. De stage.....  | 4  |
| 2.1 Locatie en duur .....   | 4  |
| 2.2 Het bedrijf.....  | 4  |
| 2.2.1 De geschiedenis.....  | 4  |
| 2.2.2 Nu .....  | 5  |
| 2.3 De opdrachten in het kort .....   | 7  |
| 2.4 De opdrachten uitgewerkt.....   | 7  |
| 2.4.1 De inwerkperiode .....  | 7  |
| 2.4.2 De hoofdfase .....  | 8  |
| 2.5 Omstandigheden.....   | 11 |
| 2.6 Wat ik heb geleerd .....  | 11 |
| 2.7 Beoordeling.....  | 12 |
| 2.7.1 Het nut van mijn werkzaamheden .....  | 12 |
| 2.7.2 De uitvoering van de opdrachten.....  | 12 |
| 2.7.3 Mijn eigen geïnitieerde werkzaamheden, mijn optreden en mijn functioneren ..... | 12 |
| 2.7.4 De kwaliteit van de begeleiding .....   | 12 |
| 2.8 Leerpunten na mijn stage .....  | 13 |
| 3. Slot .....   | 14 |



## 1. Inleiding

De eerste helft van mijn derde studiejaar ben ik bij Quadrant werkzaam geweest als stagiair (programmeur). Gedurende deze periode heb ik ervaren hoe het is om binnen een bedrijf werkzaam te zijn. Ik zat op een afdeling die onderzoek doet naar C# .NET. Zij zijn met name aan het kijken hoe ze dit kunnen inzetten in de toekomst om hun product om te bouwen of te herbouwen. Dit omdat het huidige product, genaamd King, is geschreven in Delphi en dit aardig is verouderd. Ook is dit niet Object georiënteerd geprogrammeerd, wat het onderhoud heel lastig maakt.

In dit document zal ik aan de lezer laten zien wat ik heb gedaan en geleerd om mijn stage tot een succes te maken.



## 2. De stage

### 2.1 Locatie en duur

Mijn stage heb ik gelopen bij Quadrant Software bv. Dit bedrijf bevindt zich aan de Eglantierbaan 95 in Capelle aan den IJssel. De stage was van 1 september 2010 tot en met 6 januari 2011.

### 2.2 Het bedrijf

#### 2.2.1 De geschiedenis

In 1981 richtten Hans Eschauzier, Ton Karelse en Wim Wijting, die elkaar kenden vanuit de studentenwereld, Quadrant Automatisering op. In eerste instantie een systeemhuis, dat begint met de verkoop van Data General Minicomputers, in combinatie met een boekhoudprogramma. Om te zeggen dat de zaken meteen uitstekend liepen, is enigszins bezijden de waarheid. Hans Eschauzier: "Zo'n Data General Mini, nu in onze ogen een enorm apparaat dat bijna niets kan, moest toen toch rond de 70.000 gulden opbrengen, voor één gebruiker. Dat was dan wel inclusief programmatuur en printer, maar het bleef een hoop geld. Door die hoge prijs verkochten we ze nauwelijks."

Het aanvankelijke gebrek aan succes was voor Karelse en Wijting aanleiding om in te gaan op andere aanbiedingen. Maar Hans Eschauzier ging door met Quadrant. Hij kreeg in 1982 versterking van Cees Scholten, eigenaar van een reclamestudio en de toenmalige huisbaas van Quadrant. Het bedrijf richtte zich toen op de microcomputers die rond deze tijd voorzichtig op de markt kwamen. "Wij kozen voor de Piccolo, een Deens product. Nog niet goedkoop hoor, rond de 20.000 gulden voor een apparaat met twee 8-inch diskettestations van 900 Kb. Het boekhoudprogramma voor de minicomputer draaide hier niet op. We ontwikkelden toen zelf QAS, het Quadrant Administratie Systeem."

QAS was het eerste boekhoudprogramma dat Quadrant zelf op de markt bracht. Groot succes bleef in eerste instantie uit. "We waren klein en hadden geen promotiebudget", vertelt Eschauzier. Maar vanaf 1983 kwam de personal computer op de markt en Quadrant zag direct de enorme mogelijkheden voor een standaard-boekhoudpakket, speciaal ontwikkeld voor de PC.

QAS werd aangepast voor gebruik op de PC en voor de verkoop werd een nieuwe aanpak gebruikt: de Borland-methode. Het Amerikaanse bedrijf Borland had namelijk veel succes met het tegen een zeer lage prijs aanbieden van een goed product, want door de lage prijs werden er zoveel pakketten verkocht, dat daardoor toch een hoge totale opbrengst gerealiseerd werd.

Eschauzier: "We deden dat trouwens wel via een omweg. Een vereenvoudigde versie van ons administratieprogramma werd door Computers Direct uitgebracht onder de naam QUEEN voor de extreem lage prijs van 299 gulden. En dat werd wél een doorslaand succes, want het eerste jaar werden er al 2.500 stuks van verkocht!"



Dat succes betekende voor het bedrijf een behoorlijk aantal veranderingen. In 1987 werd KING uitgebracht, een uitgebreidere versie van QUEEN. Voorheen deed Quadrant ook nog wel eens wat maatwerk op automatiseringgebied, maar deze activiteit werd langzaam maar zeker afgebouwd. Quadrant richtte zich voortaan op het ontwikkelen en verkopen van standaardsoftware. Er kwam een echte sales-afdeling, er werd een helpdesk ingericht en er groeide een dealernet dat op een gegeven moment niet minder dan 600 verkooppunten kende. Erkenning door gebruikers en deskundigen volgde. De Stichting Software Belangen kende KING niet minder dan vier sterren toe, het absolute maximum voor een financieel pakket. In 1991, bij het tienjarig bestaan van Quadrant, waren er bijna 15.000 programma's verkocht. Inmiddels was Quadrant al verhuisd van Rotterdam naar een nieuw kantoorpand in Capelle aan den IJssel.

De markt van de negentiger jaren is anders dan die van de jaren '80. Natuurlijk zijn er nog wel starters, maar vrijwel alle financiële afdelingen werken inmiddels met een boekhoudprogramma. Quadrant Automatisering onderscheidt zich in deze vervangingsmarkt vooral door de prijs/prestatieverhouding en gebruiksvriendelijkheid van de pakketten. Eschauzier: "Nog steeds is het hebben van een behoorlijke groep gebruikers de basis voor het aantrekken van nieuwe klanten. Want als je al veel klanten hebt, dan moet je product wel goed zijn. Bedrijven willen echter niet alleen een goed product, maar ook een product waar een betrouwbare organisatie achter staat. Die bieden wij ook."

Halverwege de negentiger jaren gingen de wegen van Quadrant en Cees Scholten weer uiteen. Sindsdien staat Hans Eschauzier alleen aan het roer. De jaren negentig werden gekenmerkt door toenemende professionalisering. In 1995 behaalde het bedrijf als een van de eerste softwarehuizen het ISO 9001-certificaat. Het dealernet werd geschoond en telt nu ongeveer 200 goede en enthousiaste wederverkopers. De groei zit er goed in.

Een belangrijke gebeurtenis in deze periode is de introductie van Microsoft Windows 95. Eschauzier: "KING draaide tot dan toe onder DOS, een naar onze mening uitstekend functionerend besturingsprogramma, zeker voor data-entry (het invoeren van grote hoeveelheden gegevens, zoals journaalposten). Maar Windows 95 sloeg aan. Er kwam vraag naar KING voor Windows en vanzelfsprekend gingen we aan de slag." Het werk duurde langer dan eerst was gedacht. "We moesten telkens de releasedatum aanpassen, tot verdriet van de sales-afdeling." Maar in 1999 kwam de eerste release van KING voor Windows uit, gevolgd door uitgebreidere releases in 2000 en 2001. Een belangrijke ontwikkeling, al is het wel zo dat veel boekhoudadministraties nog zonder problemen onder DOS werken. In het najaar van 2001 komt een KING-module voor het beheren en administreren van voorraden beschikbaar. Eschauzier: "Een product waaraan behoefte bestaat. We verwachten hiermee een flinke omzetgroei te kunnen realiseren."

Bron: Quadrant Jubileummagazine 2001

### 2.2.2 Nu

#### *Situatie*

Inmiddels zijn er al zo'n 40.000 King- en Queen-pakketten verkocht! Hiermee is Quadrant uitgegroeid tot nummer 2 op de markt van de financiële standaardsoftware.

De reden van dit succes is geen geheim. Het is de prijs-/prestatieverhouding van het pakket. Met andere woorden: King is een goed pakket voor een gunstige prijs. Kwaliteit staat bij Quadrant hoog in het vaandel. De totale productontwikkeling inclusief de grondige test verloopt volgens vaste procedures.



Quadrant Automatisering BV is de holding van een drietal werkmaatschappijen: Quadrant Software BV, Quadrant International BV en Handata BV. Quadrant Software ontwikkelt en verkoopt de software, verzorgt de ondersteuning van de King- en Queen-gebruikers (op basis van een serviceabonnement) en geeft King-cursussen. Quadrant International bedient het buitenland. Handata BV is gespecialiseerd in mobiele automatisering. Het totale Quadrant-team telt ruim 60 mensen.

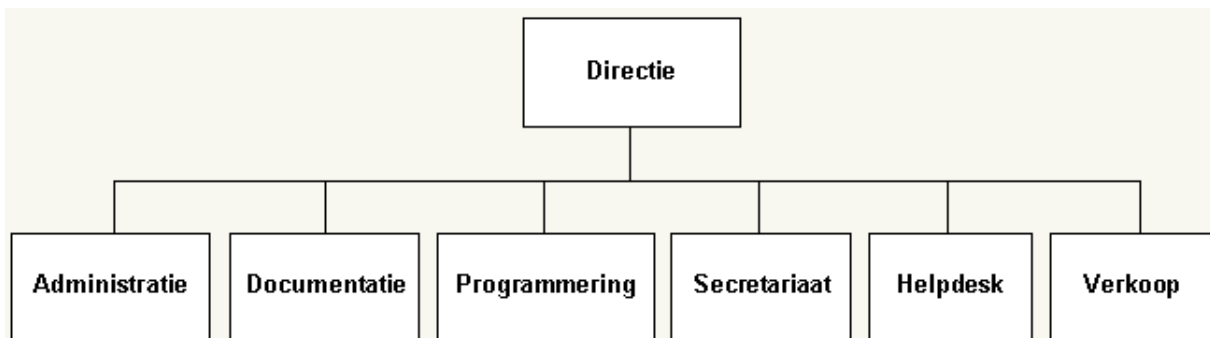
Bron: <http://www.quadrant.nl>

Financieel gaat het ook uitstekend met Quadrant. Op dit moment staan er geen leningen uit bij de banken. Dat is in deze moeilijke tijd een goede positie. Ook heeft het bedrijf een eigen wagenpark, waarvoor geen leningen uit staan.

Ten slotte is ook de omzet van 2010 al gedeeltelijk bekend. Deze overtreft zelfs de voorgaande jaren 2008 en 2009.

### *Indeling*

Het bedrijf bestaat uit 6 onderdelen, namelijk Administratie, Documentatie, Programmering, Secretariaat, Helpdesk en Verkoop. Daarboven staat de Directie.



Wat houden ze in? Hieronder staan een aantal (kern)bezigheden van de onderdelen.

#### *Directie*

Deze houdt zich bezig met het beleid inzake de ondernemingsdoelstellingen, productontwikkeling, verkoop en marketing, kwaliteit en personeel. Daarnaast houdt ze de planning bij, zorgt voor reclame in de media, betalingen en handelt klachten af. Ook beheert ze de personeelszaken en het wagenpark.

#### *Administratie*

Zij houdt zich bezig met de controle en correctie van bestellingen, orderafhandelingen en facturering, klachtenregistratie, archivering, boekhouding, loonadministratie, voorraadbeheer en nog meer onderdelen.

#### *Documentatie*

Hier wordt de gebruikersdocumentatie gemaakt, de lay-out van documenten verzorgd, de website beheerd en de kwaliteitsdocumentatie gedaan.



### Programmering

In deze afdeling wordt de programmatuur gebouwd, onderhouden en getest. Deze groep bestaat weer uit een groep die zich bezig houdt met King Mobile (voor op scanners met Windows Mobile), een groep die aan de "oude" King werkt in de taal Delphi en een groep die is overgegaan naar .NET om wat terrein te verkennen om dit over een paar jaar te gebruiken als Delphi niet meer gebruikt gaat worden. Ook valt hier het systeembeheer onder en het beheer van de telefooncentrale en het maken van back-ups.

### Secretariaat

Zij beheert de inkomende post, zorgt voor het ontvangen van gasten (dus ook de receptie), coördineert het beheer van de keuken, koffiezetapparaten, kopieerapparaten, besteld voorraden voor de keuken, beheert de agenda van de directie, geeft ziekmeldingen door en nog veel meer.

### Support

Handelt telefonische en schriftelijke support af, handelt de licentiegegevens af, maakt cursusmateriaal voor de gebruikerscursussen, voert support op locatie uit en verzorgt de dealertrainingen en -examens.

### Verkoop

Verzorgt de afhandeling van dealeraanvragen en -overeenkomsten, maakt registratie van aanvragen, telefonische verkoop van het hele assortiment, verzorgt de dealertrainingen en nog veel meer.

## 2.3 De opdrachten in het kort

Mijn bezigheden binnen Quadrant waren zeer divers. Aan het begin heb ik me ingelezen in de documentatie van het bedrijf die op intranet op een wiki staat. Hierna heb ik een aantal tutorials gedaan die daar ook stonden en een aantal boeken doorgewerkt (bijvoorbeeld C# van Head First door O'Reilly). Daarna heb ik het boekhoudpakket zelf bekeken. Hierna kwamen de diverse opdrachten die ik in het volgende punt behandel.

## 2.4 De opdrachten uitgewerkt

### 2.4.1 De inwerkperiode

De eerste twee tot drie weken heb ik me beziggehouden met het inlezen in de documentatie. Binnen Quadrant is er een wiki waarop van alles wordt bijgehouden. Hierop staat bijvoorbeeld waar men mee bezig is, wat er voor problemen zijn en hoe ze opgelost kunnen worden, maar ook handleidingen voor de afdeling support. Daarnaast ook alle agendapunten voor een vergadering en uiteraard de notulen.

Op de wiki staan ook de inwerk-instructies voor nieuwe collega's die naar de .NET omgeving omschakelen. Daarop staat ook een lijst met boeken en artikelen die handig zijn om te lezen. Hiervan heb ik het boek Head First C# helemaal behandeld. Daaruit blijkt maar weer dat C# inderdaad niet eens zo heel veel van Java afwijkt en dat ik hier vrij snel mee aan de slag zal kunnen gaan.

Daarnaast heb ik een artikel gelezen over SOLID. In dit artikel kwam het principe van "One Step at a Time" naar voren. Software kan aan het begin snel (misschien ook wel slordig) zijn gemaakt, omdat het maar voor één bepaald persoon of doel is gebouwd. Echter kan dit programma zo'n succes zijn dat je het wilt gaan uitbreiden door meer functionaliteit toe te voegen. Dit houdt dus in dat je vanaf het begin moet gaan nadenken hoe je je software hebt ontworpen en als je het gaat uitbreiden, wat je allemaal moet aanpassen. Je gaat dan bijvoorbeeld algemenere dingen verplaatsen naar een interface.



Hierna ben ik bezig geweest om mijn computer in te richten zodat ik aan de slag kon gaan. Dit houdt in dat ik bijvoorbeeld Visual Studio heb geïnstalleerd, maar ook de plug-ins voor Subversion, Resharper om de opmaak van de code te waarborgen, Sybase SQL Anywhere en vele andere tools.

Behalve boeken, vind je op de wiki ook tutorials. Deze gaan over de meest uiteenlopende, maar wel belangrijke zaken die je wel moet kennen. Er is bijvoorbeeld een tutorial over Subversion, maar ook over NUnit testen, WPF, XAML, LINQ, NHibernate en nog veel meer. Al de tutorials heb ik gedaan. Hierdoor heb ik een beter idee wat er in de ontwikkelgroep allemaal gebruikt wordt en hoe zij het doen.

Elke woensdag is er een kennisoverdrachtsessie (mits er voldoende agendapunten zijn). Op deze vergadering worden er allerlei nieuwe dingen die gebruikt worden uitgelegd of gepresenteerd. Daarnaast worden er ook problemen en vragen die er spelen besproken en kunnen we met elkaar zoeken naar een oplossing. Op de woensdag in de tweede week heb ik dit voor het eerst meegemaakt. Omdat ik ook als stageleerdoel had geformuleerd dat ik wil oefenen met vergaderen/notuleren, mocht ik vanaf nu al deze bijeenkomsten notuleren. Ik moet zeggen dat dit mij erg veel heeft geholpen en dat ik nu wel weet hoe je het notuleren makkelijk aan kunt pakken.

De tutorials waren gebouwd in Visual Studio 2008. Dat terwijl de afdeling over is gegaan naar Visual Studio 2010. Als je daarom een tutorial opende, vroeg Visual Studio of je het wilde converteren. Dat heb ik elke keer gedaan en dan ook weer in de nieuwe indeling teruggestopt in de Subversion tak. Daarnaast waren er een aantal referenties naar .dll bestanden met een vast versie nummer. Maar deze tools waren ondertussen ook al verder ontwikkeld en er trad dan ook een referentie probleem op. Ik heb tegen die referentie gezegd dat het versienummer niet van belang is en ook dat weer met Subversion gecommitt. Dit was eigenlijk mijn eerste echte bezigheid in Subversion, waarin ook de software zit die wordt ontwikkeld.

#### 2.4.2 De hoofdfase

Op 14 september (de dinsdag in mijn 3<sup>e</sup> stageweek) heb ik Quadrant.NET uit Subversion gehaald en ben ik echt begonnen. Allereerst uiteraard even de globale opzet bekijken. Hierna heb ik even een aantal simpele classes van naam veranderd (refactoren). Hiermee heb ik gelijk mijn eerste leerpunt te pakken. De volgorde van handelen: Eerst al je wijzigingen maken, dan je project builden en controleren op errors. Daarna al de unittests runnen en als al deze punten goed zijn gegaan, kun je het committen naar Subversion. Dit zal opgemerkt worden door de TeamCity CI (Continuous Integration) server, die alle veranderingen uit Subversion haalt en zelf ook zal gaan builden en testen. Als ook dit slaagt, heb je alles goed gedaan en zijn er verder geen (bekende) problemen opgetreden.

De volgende dag kreeg ik mijn eerste echte opdracht. In het offerteprogramma waar we momenteel mee bezig zijn, kun je offertes invoeren met offerteregels. Deze voer je in door het artikelnummer (of zoekcode) op te geven, waarna er een aantal andere velden als prijs en tekst op offerte worden ingevuld. Dit laatste was echter nog niet geïmplementeerd en zou ik moeten doen. De database er omheen was er al wel en de teksten stonden daar ook al in. Er ontbrak alleen nog een entity en een mapping voor NHibernate naar de database.

Ik ben begonnen met het maken van de entity. Dit houdt niet meer in dan een class met daarin de attributen die uit de database moeten komen. Dus een entity is in principe de in code gebouwde databaseregel. In dit geval bevat de entity een Taal en de Tekst.

De situatie is als volgt: Je hebt een OfferteEntity. Deze bevat OfferteRegelEntities. Hierin zitten weer Artikelen.





In ArtikelEntity wordt door middel van een mapping alle teksten (nu nog in alle talen) opgehaald voor dit artikel. In de offerte wordt bepaald welke taal de offerte zal handhaven. Dit weet hij door de taal van de ingevoerde debiteur op te vragen. Het kan echter zijn dat er geen debiteur is ingevoerd. In dat geval zal de basistaal van de administratie gekozen worden. Zodra je nu een nieuwe offerteregel toevoegt, wordt deze taal doorgegeven en weet de regel het ook. Als je nu in die regel een artikelnummer (of zoekcode) invoert, wordt (weer door een mapping) de juiste artikeltekst (met de juiste taal), uit het artikel gehaald en weergegeven.

Met deze opdracht ben ik ongeveer tweeënhalve week bezig geweest. Dit is op zich vrij lang. Dat komt omdat ik nog heel de structuur van de bestaande code moest doorkrijgen en ook moest leren wat NHibernate nu eigenlijk is en wat het doet en hoe zijn mappings werken. Als ik deze opdracht weer zou moeten doen, verwacht ik dat ik er niet meer dan een week mee bezig ben.

Binnen Quadrant wordt ook om de andere week een afdelingsvergadering gehouden. Hier ben ik ook elke keer bij aanwezig geweest. Deze vergadering is wat breder dan de kennisoverdrachtsessies die ik eerder heb uitgelegd. Hierin worden bijvoorbeeld huishoudelijke zaken behandeld en vragen/problemen die het hele bedrijf/ de hele afdeling aangaan besproken. Ook hier komen af en toe demo's in voor van de projecten die er binnen het bedrijf lopen.

Na de eerste opdracht was het weer tijd voor verdieping. Op internet was een presentatie te vinden over Value Objects<sup>1</sup> (binnen Quadrant valuetypes genoemd). Wat zijn valuetypes? Dat zijn classes met methoden en operators (een operator is bijvoorbeeld plus, min, keer, enz. maar ook de == constructie waarmee je twee objecten kunt vergelijken) die bij elkaar horen. Zo heb je software waarin een telefoonnummer als een string is gedeclareerd en daarna op diverse plaatsen in hun code gecontroleerd wordt of het daadwerkelijk een telefoonnummer is. Als je dit oplost door een valuetype Telefoonnummer aanmaakt, voorkom je dat ergens een onmogelijk telefoonnummer voorkomt. Intern is het valuetype nog steeds wel een string, maar als je een nieuw telefoonnummer aanmaakt, wordt er eerst gecontroleerd of het een valide nummer is. Zo niet, dan heb je al bij de invoer een error en niet ergens later in je code.

Daarnaast heb ik geleerd wat het inhoud om unittests te schrijven. Op school is hier eigenlijk nooit aandacht aan besteed. Achteraf stelt het misschien niet zo veel voor, maar het is toch handig om dat even onder de aandacht te brengen. Voorheen zou ik een test hebben geschreven die alles op allerlei manieren zal testen. Dit is echter overdreven. Je moet alleen code testen waar je problemen verwacht, of code die meer doet dan alleen een getalletje of tekstje teruggeven.

Nadat ik dit heb geleerd, heb ik het direct in de praktijk gebracht. We hebben hier een heleboel valuetypes die nodig eens opgeschoond moeten worden en waarvoor nog geen tests geschreven zijn. Hier ben ik dan ook ongeveer vier weken mee bezig geweest en ik kreeg zelfs complimenten dat ik het testen zo goed doorhad!

Op de woensdag in de 10<sup>e</sup> week van mijn stage heb ik weer een grotere opdracht gekregen. NHibernate laadt de mappings in het programma om het te kunnen gebruiken. Elke keer dat je het programma opstart, wordt dit uitgevoerd. Dit is niet efficiënt omdat dit vrij lang duurt. In de code is al een switch gemaakt om deze cache naar een bestand op te slaan om het daarna weer in te kunnen laden. Deze stond echter uit omdat er een aantal problemen mee optraden. Eén van de problemen was dat wanneer je zelf iets aan de code had aangepast, de cache verwijderd dient te worden. Dit kon echter niet helemaal door C# worden bepaald. Hiervoor moest ik iets bedenken dat dat wel kon.

---

<sup>1</sup> Value Objects van Dan Bergh Johnsson.



Allereerst heb ik dit geprobeerd door te kijken of je van een assembly (het project dat is gebuild door Visual Studio als een .dll bestand) een hashcode kan opvragen en die met de volgende run te vergelijken. Echter wordt bij het compileren altijd ergens iets aangepast door Visual Studio, zodat deze vergelijking altijd zal denken dat de assembly anders is.

Na een week heb ik dit opgegeven en ben ik een ander idee gaan uitwerken. We maken allemaal gebruik van Subversion. Daarin zit versiebeheer. Subversion “weet” dan ook of een bestand lokaal is veranderd, maar ook, als het niet veranderd is, welk versienummer het heeft, zodat het ook weet of er extern wijzigingen zijn gemaakt. Dit is precies wat ik zocht!

Ik heb een programma gemaakt dat de te onderzoeken paden uit een XML bestand inleest en dan via de commandline vraagt of er op dat pad wijzigingen zijn aangebracht. Dit gebeurt via het commando “svn diff <pad>”. Als er wijzigingen zijn, zullen de cache bestanden van NHibernate, die ook in de XML staan, worden verwijderd, zodat het programma ze weer opnieuw kan opbouwen. Om ervoor te zorgen dat nieuwe wijzigingen, als er geen commit is gedaan, toch worden gezien, sla ik de Subversion resultaten op en vergelijk die ook met elkaar. Het CacheCheck programma kan dus wel zeggen dat Subversion wijzigingen ziet, maar als de resultaten met elkaar overeenkomen, zijn er geen nieuwe wijzigingen gemaakt sinds de vorige keer en hoeft de cache niet opnieuw weggegooid te worden.

Ook heb ik onderzoek gedaan naar het gebruik van certificaten, zodat je software beveiligd is tegen hackers. Dit certificaat laat zien dat de software afkomstig is van een erkend bedrijf en dat er niet mee geknoeid is nadat het is gecertificeerd. King bestaat uit een setup.exe Deze zal weer een aantal kleinere setups uitvoeren. Het zou dus kunnen dat het betreffende setup bestand wordt vervangen door een aangepaste versie. Dit is dan ook niet de bedoeling. Ik moest uitzoeken of het mogelijk is om in de setup (opgebouwd door middel van C++ code) het certificaat van de andere setup bestanden op te halen en uit te lezen is. Dit is mij gelukt: Ik kon de naam van de leverancier en de thumbprint (unieke code die elk bedrijf krijgt bij een certificaat) ophalen en dus ook vergelijken of het de correcte gegevens zijn.

Verder heb ik tussendoor nog een heleboel kleine dingen gedaan. Bijvoorbeeld het oplossen van fouten in unittests. Er zat bijvoorbeeld een test tussen die altijd goed ging. Echter niet toen de switch werd aangezet die de cache naar een bestandje opslaat en het later weer inleest. Het bleek dat in de mapping een verkeerd type werd gebruikt. Er stond een Int32, maar in de entity was het een long. Dit ging alleen fout als de cache was ingeladen, want die klopte niet. Dit heb ik vrij snel kunnen oplossen en nu werkt het zoals het hoort. Het bleek zelfs een fout te zijn die al langer bekend was, maar waar nog nooit naar is gekeken.

Zo was er nog een test, die maar heel af en toe, op de raarste momenten fout ging. Deed je hem opnieuw, was hij weer goed. Deed je hem nog een keer, dan kon het weer fout zijn. Het ging om een test dat een lijstje met items aanmaakte, dat opsloeg en weer opnieuw inlas. Echter bleek de volgorde niet meer te kloppen. Ook dit is opgelost in de mapping door een “order-by” statement toe te voegen. Dit heb ik samen met een andere collega gedaan en we zijn er dus gelukkig uitgekomen.

Als laatste opdracht heb ik een aantal testdatabases opgeruimd. Officieel is er één testdatabase. Echter zijn er in de loop van de tijd een aantal bijgekomen om op specifieke delen te testen. Dit is echter erg omslachtig. Daarom zijn de meeste tests verhuisd naar de originele testdatabase. Dit houdt in dat je de desbetreffende unittests moet omzetten naar de originele database en, indien nodig, aanpast zodat ze weer werken. Ik heb drie aardig grote databases kunnen omzetten.



## 2.5 Omstandigheden

Ik heb op een afdeling gezeten dat onderzoek doet naar .NET. Deze afdeling bestaat nu uit zeven programmeurs, met mij meegerekend. Het is een groep die goed met elkaar overweg kan en als er problemen zijn elkaar te hulp schiet. Dit werkte dus ook voor mij. Als ik een vraag had, kon ik die gewoon stellen en dan werd ik ook geholpen als dat nodig was.

Ook was het werk wat ik had zeker gevarieerd. Zo heb ik grotere opdrachten uitgevoerd en dan ook tussendoor nog probleempjes op lossen en daarnaast nog een aantal wat kleinere opdrachtjes. Ik had dus ook genoeg te doen en als dat niet zo was, dan gaf ik dat aan en werd er weer iets nieuws bedacht. Er lagen nog een heleboel punten open die nog gedaan moesten worden, dus er kwam altijd wel weer wat nieuws uit!

Het laatste programmaatje dat ik heb gebouwd, werd met enthousiasme ingezet in het project. Het boekte resultaten, want daarna startte het project in 4 in plaats van 6 seconden op. Dit lijkt niet veel, maar het is wel een versnelling van 33 procent! Deze resultaten en de gedachten hoe ik het programma heb gemaakt, heb ik op een kennisoverdracht sessie "gepresenteerd". De reacties die ik kreeg waren in ieder geval positief. Dit motiveert uiteindelijk ook je stage en je hebt er plezier van. Je beseft dat je het dan in ieder geval niet voor niets doet en het ook daadwerkelijk ingezet wordt.

Als we dan gaan kijken naar wat ik op school allemaal geleerd heb, mis ik wel het een en ander. Op school heb ik gewerkt met Java. Verder is er niet geleerd dat er nog andere talen zijn en wat hun voor- en nadelen zijn. Ik zelf vind dus dat er teveel is gefocust op één taal. Er zouden bijvoorbeeld een aantal keuzevakken kunnen komen die andere talen aanleren. Of misschien moet dit ook, al is het maar de basis, een verplicht vak worden. Je leert dan ook gelijk met een andere editor werken, in plaats van alleen maar NetBeans.

Ook mis ik uitleg over het unittesten. In project78 moesten we een aantal unitests meeleveren. We hadden echter helemaal geen idee wat het was en hoe het moest. Ons werd verteld om dit op internet op te gaan zoeken. Ja, dan weet je wel wat een unittest doet en hoe je het schrijft, maar niet precies wat je wel en niet moet testen en op welke manier.

Wel heb ik steun gehad aan de projecten die ik op school heb gevolgd en de vakken die te maken hebben met programmeren. Hier heb ik geleerd om object georiënteerd te programmeren, wat het betekent als je iets static maakt en wat er bedoeld wordt als iets public, private of protected is. Dit betreft de praktische vakken. Ook heb ik veel gehad aan de modules van business skills, omdat je dat in elk bedrijf wel weer terug ziet. Zo heb ik bijvoorbeeld de vergaderingen mogen notuleren.

## 2.6 Wat ik heb geleerd

Voordat mijn stage begon, heb ik voor mijzelf drie duidelijke leerdoelen opgesteld.

Ten eerste wilde ik kennis maken met een andere ontwikkeltaal, in dit geval C# (in combinatie met .NET). Dit omdat ik vond dat het te risicovol is om maar één taal te kennen, omdat talen snel verouderen en zelfs weg kunnen vallen. Ook verbreedt het je kennis en dat levert weer meer kansen op op de arbeidsmarkt. Voor mijzelf had ik het doel om deze taal bijna net zo te beheersen als dat ik Java beheers. Dat wil zeggen dat ik zonder al te veel moeite de basis functies kan schrijven. Dit is mij goed gelukt. Mede omdat de overeenkomst met Java best groot is. Natuurlijk zijn er altijd kleine verschillen, zoals Java gebruikt "imports" om externe libraries te koppelen en C# doet dat met "usings".

Ten tweede wilde ik mijn notuleer prestaties verbeteren. Ik wilde eenvoudig kunnen samenvatten wat er allemaal werd besproken. Dit heb ik aardig onder de knie kunnen krijgen, omdat ik elke kennisoverdracht sessie mocht notuleren. Ik heb dus flink kunnen oefenen.



Ten slotte had ik nog een klein derde puntje. Ik wilde ervaren hoe het is om (bijna) fulltime in een bedrijf te werken. Dit leer je natuurlijk vanzelf tijdens je stage die vier dagen in de week duurt. Het is inderdaad even wennen, omdat je aan vaste tijden van 's ochtends tot 's avonds aan het werk bent, terwijl je op school een heel divers rooster hebt en misschien zelfs een dag vrij hebt. Dit went allemaal echter vanzelf en na verloop van tijd vind je dit niet eens meer erg.

## 2.7 Beoordeling

### 2.7.1 Het nut van mijn werkzaamheden

De opdrachten die ik heb gedaan, hebben zeker nut in het uiteindelijke product. Bijvoorbeeld het maken van de unittests, die waarborgen de werking van de code. Als er ergens anders iets wordt aangepast aan de code en de test faalt, dan weet je dat er iets aan de situatie is veranderd wat niet de bedoeling is.

Ook het opruimen van de valuetypes is blijvend. In de valuetypes waren een aantal methoden niet helemaal in orde of werden niet meer gebruikt. Dit heb ik er uit gehaald of aangepast en dit komt dus uiteindelijk ook zo in het product.

Daarnaast heeft het programma dat ik heb gebouwd om de cache te verwijderen, ook nog voordeel: Het project start nu namelijk 2 seconden sneller op. Hiermee is de ontwikkel cyclus dus iets versneld, want er hoeft niet bij iedere rebuild een nieuwe cache aangemaakt te worden.

### 2.7.2 De uitvoering van de opdrachten

Het uitvoeren van de opdrachten ging af en toe moeizaam, omdat ik dan weer wat moest uitzoeken hoe het werkt. Met name de eerste weken heb ik veel moeten nalezen over C# en wat het een en ander betekend. Na verloop van tijd ging het echter steeds beter en ik vind dat ik nu wel aardig vertrouwd ben geraakt met C#. Daarom ging de opdracht van de cache ook sneller en eenvoudiger qua programmeren. Hier liep ik echter weer tegen het probleem aan dat ik moest uitzoeken hoe je met C# de commandline van Windows aanroept. Ook moest ik weten wat bepaalde Subversion commando's deden en hoe ik ze kon gebruiken. Het programma maakt ook gebruik van XML. Dus ook dat moest ik weer even bekijken, hoe je dat met C# kan uitlezen.

Al met al vind ik dat ik de opdrachten die ik heb gekregen met succes heb kunnen maken en dat ik er nog aardig wat van heb geleerd ook.

### 2.7.3 Mijn eigen geïnitieerde werkzaamheden, mijn optreden en mijn functioneren

Het opschonen van valuetypes en het oplossen van andere problemen die ik zelf tegen kwam, waren geen opdrachten die ik moest doen. Ik heb het echter wel aangepakt, omdat het nodig was om te doen. Alle punten die gedaan moeten worden staan op een backlog op de wiki. Hieruit kun je dus kiezen wat je gaat doen. Zelf vond ik dit bijzonder prettig, omdat ik dan in ieder geval, wachtend op een nieuwe opdracht, gewoon verder kon werken.

Verder vind ik mijn optreden binnen het bedrijf goed. Ik kon goed overweg met de collega's, dus ging er iets niet helemaal goed, of wist ik iets niet, kon ik bij ze terecht. Op het moment dat ik niks meer te doen had, gaf ik dat aan. Zodoende heb ik me niet zitten vervelen.

### 2.7.4 De kwaliteit van de begeleiding

De begeleiding vanuit het bedrijf is ook uitstekend. Als je wat hebt kun je bij hem terecht en dan kan hij je ergens mee op weg helpen. Als hij er niet is, wordt het opgevangen door de projectleider, waarbij je ook goed terecht kunt. Verder is het hele team behulpzaam, dus kan je vragen ook bij hen kwijt.

De begeleiding vanuit school beoordeel ik met een voldoende/goed. Als ik hem nodig had, dan kon ik bij hem terecht. Verder hebben we elkaar niet echt nodig gehad, dus kan ik hier niet veel over zeggen.



## 2.8 Leerpunten na mijn stage

Wat ik zeker wil verbeteren, is het om hulp vragen als iets niet helemaal lukt. Nu heb ik eigenlijk zoveel mogelijk zelf proberen uit te zoeken. Dit komt met name omdat ik onbekend was in deze taal. Nu ik dit echter door heb, kan ik bij problemen beter eerder om hulp vragen. Al is het maar dat ik weet hoe een ander er over denkt, of ik in de juiste richting aan het zoeken ben of dat iemand anders een beter idee heeft.



### 3. Slot

Al met al was het een goede stageperiode. Tijdens mijn stage heb ik genoeg te doen gehad en ik heb nu ook zeker een duidelijk beeld hoe het er in het bedrijfsleven aan toe gaat. Daarnaast heb ik ook veel geleerd van het programmeren zelf en ik weet dat er nog veel meer te leren valt. Dit komt allemaal echter vanzelf, want dat past niet allemaal in de relatief korte stageperiode. Ook de sfeer onderling was wat mij betreft uitstekend.