



## **Sigma**

Geschiedenis van een softwaretester

Jens Pas

Dit is geen artikel. Dit is ook geen column. Dit is een geschiedenis<sup>1</sup>. Een geschiedenis over een softwaretester die in een nieuw informaticaproject wordt geplaatst. De tester heet Kenneth. Het project heet Sigma. De hoofdstukken geschreven in een mengeling van Nederlands en Engels, wat gebruikelijk is voor de wereld van Kenneth.

Het gaat over project management en softwarekwaliteit, maar toch ook niet. En misschien een beetje over liefde, maar dat is niet het juiste woord. En als je er geen naam voor hebt, dan gebruik je daar een geschiedenis of legende voor. Zo werkt dat. Al eeuwen lang. Vandaar, de geschiedenis van Kenneth.

Iedere geschiedenis heeft zijn muziek zou Alessandro Baricco zeggen. Dit verhaal ook. Het vertaalt de sfeer van het verhaal. De muziek van dit verhaal is snel maar niet te hard. Geen techno, geen dance, maar wel elektrisch. Als je goed luistert, hoor je de gedrevenheid van de programmeurs. En de onzekerheid en het verdriet van de testers. De frustraties van de projectleiders. Maar ook van de “eindgebruikers”.

Geen moraal, geen boodschap, tenzij u er zelf één in legt. Wie weet.

Zo, het canvas is getekend. We kunnen beginnen. We zoeken in op de hoofdstad, de place of the action. Kenneth zit sinds vandaag op de derde verdieping van een kantoorgebouw op een nieuwe werkplek. Een “werkeiland”.

---

<sup>1</sup> Geïnspireerd op “Zijde” van Alessandro Baricco, Italiaanse schrijver – Turijn. Schreef het schitterende “Zijde”, waarop dit verhaal een beetje geïnspireerd is.



**Sigma**

**Sigma**

## 1

Hoewel Kenneth voor zichzelf ooit bedacht dat hij een groots softwareprogrammeur zou worden, had de werkelijkheid hem gemanoeuvreerd in de functie van softwaretester. Toeval eigenlijk. Kenneth was ingenieur van opleiding, 28 jaar, met een voorliefde voor computers. Thuis had hij als eerste er één gehad. Een PC, een “kloon”.

Na zijn studies had hij gesolliciteerd bij een bedrijf in Gent dat grafische software bouwde. Dat leek hem wel wat. Software die mooie beelden kon maken.

Hij startte echter niet als programmeur, maar als lesgever. Hij reisde de wereld rond, klanten uitleg geven over en helpen bij het gebruiken van die heel complexe functies. In die tijd leerde hij veel. Dingen waarvan hij niet wist dat ze bestonden. Zo gaat dat immers altijd.

Twee jaar had hij dat werk gedaan. Hij had Duitsland gezien, in Marburg gewerkt, prachtig universiteitsstadje. In Denemarken, in Bilund, waar hij leerde dat men daar om zeven uur 's ochtends begon te werken en om half vier stopte. Even in de VS en dan vooral in Frankrijk, in Nice, waar hij meerdere maanden verbleef. Daar mocht hij eindelijk eens programmeren. De klant had immers gevraagd om enkele interfaces te schrijven om de software met eigen drukmachines te laten communiceren. Kenneth's bedrijf had toen net geen programmeurs beschikbaar en hij mocht het dan zelf eens proberen. Het was alsof hij een promotie kreeg. En dan nog in Nice. Het leek wel betaalde vakantie.'s Avonds

slenteren op de Promenade des Anglais, eten in Le Vieux Nice of iets verderop in Saint-Paul de Vence of Biotte.

Maar we dwalen af. Kenneth werkt immers niet meer in dat Gents bedrijf. Na twee jaar vertrekt Kenneth uit Gent op zoek naar iets nieuws. Hij vindt een bedrijf dat software maakt om andere software te testen, een soort software-testrobotten.

Hij heeft een goed gesprek en na enkele weken is hij daar aan de slag als “testconsultant”. Kenneth is dan 26 jaar. Hij begrijpt het niet. Hij heeft nog nooit software getest, heeft twee jaar ervaring als lesgever in grafische software, heeft enkele weken op eigen houtje mogen programmeren en is nu “Adviseur”. Het staat op zijn kaartje: Consultant. Dus moet het waar zijn. Hij leert al gauw dat hij de klanten van zijn nieuw bedrijf gewoon moet uitleggen hoe ze met een soort testsoftware moeten werken. Eigenlijk lesgever. Maar dat heet nu Consultant. Kenneth dacht dat dat een titel was voor zeer ervaren mensen.

Kenneth werkt anderhalf jaar als “Adviseur”, na zes maanden zelfs “Application Specialist” bij dat Amerikaans bedrijf dat die testtool bouwt. Alles is relatief. Als het even wat minder gaat met de verkoop van de tools, ruilt Kenneth zijn titel van Specialist met die van “softwaretester”. Zijn klant, waar hij laatst advies gaf, werft hem aan. Na enkele maanden testen van updates en patches, wordt Kenneth, nu 28 jaar, in het Sigma-project gedropt.

## 2

Sigma heeft een tweeslachtige reputatie. Het is het nieuwste prestigeproject van zijn bedrijf, sterk gesponsord door de CEO maar het is tegelijkertijd een kerkhof. Meerdere projectleiders hebben er hun tanden op gebroken. Sommigen werden ontslagen, anderen verdwenen in “Special projects”. We zijn toe aan de vierde projectleider. Meerdere programmeurs, meestal de beste, hebben ontslag genomen. Enkele analisten ook, voor zover ze niet via de job van projectleider zijn vertrokken.

Sigma is een contactmanager. Een programma waarmee je adresgegevens van klanten kan bijhouden. Maar dan veel meer. Hele historieken kunnen worden bijgehouden en allerlei rapporten kunnen getrokken worden. Het bijzondere aan Sigma is dat het een “webapplicatie” is. Een programma dat je als een website gebruikt. Waar de klanten zelf hun adresgegevens kunnen aanpassen zodat je steeds de juiste gegevens hebt. De foutmarge op de juistheid van de contacten zou minimaal zijn. Dat was het verkoopsargument van Sigma. En dat Sigma kan synchroniseren met allerlei perifere toestellen en softwarepakketten, zoals Outlook of Notes of Palm, PocketPC's en zeker de nieuwe Smartphones.

De CEO zegt dat geen enkel bedrijf tevreden is met hun huidige manier van contactbeheer. Elk bedrijf worstelt ieder jaar opnieuw met het samenstellen van de adreslijsten voor de kerstkaarten, en de CEO kan het weten. Elk bedrijf stelt vast dat adresgegevens verouderd zijn en/of dat verschillende mensen in het bedrijf verschillende gegevens blijken te

hebben. Sigma lost dat allemaal op. De ultieme kerstkaartentool dus. En daar wordt de toekomst van het bedrijf op gebouwd...en de CEO kan het weten.

### 3

Op de testafdeling waar Kenneth nu zit, werken nog vier collega's.

Er is Daan, die is er het langst. Hij test al vier jaar, is 30 jaar. En zit sinds dag één op Sigma. Daan doet het goed. Hij komt fluitend werken, doet ogenschijnlijk een goeie job, want hij wordt door iedereen gewaardeerd en zelfs een beetje als referentie beschouwd. Hij is alleen wat stilletjes. Geen gangmaker. Dat heeft er trouwens voor gezorgd dat Lieve, en niet hij, de leiding van het testteam heeft gekregen. Even was Daan misnoegd. Maar hij besepte al snel dat hij zich hierdoor nog meer kon nestelen in zijn hoek van het werkeiland.

Lieve werkt ook al ongeveer vier jaar bij Beetleware (dat is het bedrijf waar Kenneth dus werkt), is iets ouder, assertief, weet van aanpakken maar speelt toch niet de grote Leider. Lieve coacht meer dan leiden. Ze is wel het gezicht van de afdeling zonder het boegbeeld te spelen.

En dan is er nog Kurt en Miriam. Kurt is de jongste, werd aangeworven als programmeur, maar moest uiteindelijk, tegen zijn zin, testen. Gevolg: Kurt komt meestal met tegenzin werken en zit meer te programmeren aan allerlei testscripts dan hij echt op zoek gaat naar fouten.

Miriam is even oud als Kenneth. Heeft weinig testervaring, maar des te meer projectervaring. Miriam werkt nog maar sinds enkele maanden in het bedrijf. Ze werd in functie van het Sigma-project aangeworven. Miriam heeft zowel een informatica als een economische opleiding, wat

maakt dat ze in het team het best de business-behoefte in de software-requirements kan detecteren. Niet onbelangrijk, Miriam is beeldschoon, zeker als je als norm het effect op de collega's neemt.

Een voorbeeld: de eerste werkdag moest er voor Miriam de nodige apparatuur worden geïnstalleerd (pc, netwerkverbinding etc.). De EDP-afdeling, die zijn reputatie van inefficiënte afdeling steeds opnieuw bevestigt, klopte alle record om in geen tijd een splinternieuwe pc te installeren, met flatscreen (het eerste in de afdeling). Accounts werden in een oogwenk aangemaakt. Kleine problemen werden direct opgelost. Na 3 uur was Miriam volledig operationeel. Het was zelfs zo erg dat de collega-testers later die week gebruik maakten van haar mailaccount om een infrastructuurprobleem te melden aan EDP, omdat ze dan direct werden bediend – in tegenstelling tot de gebruikelijke twee dagen wachttijd. Miriam dus.

Niet alleen EDP buigt nederig, ook alle programmeurs. Die kunnen de feedback van de testers niet altijd smaken, maar kijken nu toch dagelijks reikhalzend uit naar een nieuwe lading kritiek, liefst ingewikkelde, zodat Miriam ze persoonlijk moet komen toelichten.

Aan dit team werd Kenneth toegevoegd. Miriam zou Kenneth inwijden in het project.

## 4

Sigma is al elf maanden over tijd. Het is nu januari en het had in maart vorig jaar in gebruik moeten genomen worden. Maar dat kon niet. Er was iets mis met het platform. Sigma was klaar, maar de servers werkten niet naar behoren. Althans, dat is toch wat men zegt.

De nieuwste software werd gebruikt om Sigma te maken. Om het doorbraakgehalte van Sigma te verzekeren. Kenneth herinnert zich nog hoe hij onder de indruk was toen het hoofd van ontwikkeling hem vertelde dat ze zelfs een speciale overeenkomst hadden met de leverancier van het Operating System. Als eerste mocht Beetleware bèta en andere versies van de software gebruiken.

Die bèta-versies werkten natuurlijk niet zo goed. Maar dat is de tol die je betaalt om nieuw te zijn. Elf maanden.

Wat er nu zo nieuw was aan die bèta's, dat kon het Hoofd niet zeggen. Dat was te technisch voor zijn niveau. Maar het bood op zijn minst een waaier van nieuwe mogelijkheden...of zoiets.

## 5

De eerste dagen verliepen rustig. Kenneth kreeg zijn plek op het eiland, een beetje schuin voor Miriam, naast Kurt, recht tegenover Daan. De werkplek was eigenlijk maar bedoeld voor drie eilandbewoners, maar met wat geschuif, kon Kenneth er wel bij. Lieve zat als teamleader verderop.

Het duurde even vooraleer Kenneth een pc had. De voorraad was immers op. Het bedrijf was zich nog aan het beraden of er nieuwe toestellen mochten aangekocht worden. Er was in het kader van besparingen een aankoopstop aangekondigd. De logica dat elke tester toch per definitie een pc nodig heeft, was toch geen evidentie. Lieve's baas, had immers vastgesteld dat de testers niet allemaal op hetzelfde moment aan het testen waren. Dus konden de vier testers toch gemakkelijk met drie pc's werken. Er was er altijd wel eentje die vrij was...Managementlogica. Zeker en vast geïnspireerd op een model van een efficiëntieconsultant.

In afwachting van zijn pc, las Kenneth dossiers. Allerlei afdrukken van rapporten, requirementsbeschrijvingen, analyses. Het duurde al een dag om de meest recente versie te vinden.

Nu en dan toonde Miriam op haar pc hoe het testwerk in elkaar zat. Met een dodelijke nauwkeurigheid toonde ze de fouten in het programma. Niet triomfantelijk, gewoon neutraal maar loepzuiver, ontdaan van elke emotie. Haar enige emotionele reactie was dat ze het niet goed begreep waarom de programmeurs die fouten zelf niet vonden. Het was toch zo

evident. Bij sommige programmeurs had ze toch twijfels over hun capaciteiten.

Maar zelfs die emotionele reactie was nogal kaal, niet mooi ingekleed wat haaks stond op haar eigen verpakking. Diplomatie was een woord dat niet direct in haar woordenboek voorkwam. To the point en direct. Haar uiterlijk maakte alleen maar dat de opmerking nog harder aankwam bij de betrokkene. En ze kon met haar brede ontwapenende glimlach elke spanning ontzenuwen. Nog nooit had ze een conflict gehad met de programmeurs, zelfs niet met de teamleaders.

## 6

Lieve kwam op het einde van dag weer eens langs bij Kenneth. Bleek dat de dossiers die Kenneth bestudeerde niet echt relevant waren. Ze dateerden immers nog van toen het team nog vol goede moed hun werk probeerde te documenteren. Nu, met die elf maanden achterstand, had men die aanpak al lang laten vallen. Nu programmeerde men vooral op basis van flarden e-mails en mondelinge overlevering. Het leek te werken.

Alleen voor de testers was het een frustrerende bedoening. Vaak vergat men hen te mailen en bijna nooit waren ze aanwezig op besprekingen. Van mondelinge overlevering was bij hen dus al geen sprake. Niet zelden werden testen gemaakt op basis van gezond verstand en persoonlijke interpretaties. En daar was eigenlijk niets mis mee. Ook deze testen vonden veel fouten, en daar was het uiteraard om te doen.

Tijdens de wekelijkse opvolgingsvergaderingen liep het fout. Bij het overlopen van de fouten met de teamleader van de ontwikkelaars en het Hoofd, kwam steeds als opmerking terug dat de testers geen gedocumenteerde baseline hadden, geen beschrijving van wat het resultaat eigenlijk wel had moeten zijn. Veel fouten konden dus niet objectief worden aanvaard. Ze waren wellicht het gevolg van het gemuggenzift van Kurt en Miriam. Kurt werkte toch tegen zijn zin, dus zijn opmerkingen waren eigenlijk zijn persoonlijke frustratie. Want bij Daan liep het allemaal goed. En Daan was er al zolang, dus hij was toch de referentie.

Daarenboven waren de fouten slecht gedocumenteerd waardoor ze moeilijk gereproduceerd konden worden. Volgens de teamleader van de programmeurs was dat dan ook de reden waarom zijn team zoveel achterstand opliep. Ze moesten veel te veel tijd spenderen aan het bestuderen van de gerapporteerde fouten. De testers verstoorden eigenlijk hun werk en concentratie.

## 7

Na drie dagen kwam er dan toch een pc voor Kenneth.

De secretaresse van de CEO had net een nieuw toestel gekregen. Wat een geluk. Kenneth kon aan de slag met het tweedehands toestel van de secretaresse. Alleen, hij kon niet tegelijkertijd testen en mails sturen, want daarvoor was dat toestel niet krachtig genoeg. Maar bon, zo erg kon dat niet zijn. Hij kon altijd zijn bevindingen neerschrijven of zo en dan achteraf via mail doorsturen. Dat zou trouwens wellicht een meer efficiënte aanpak zijn dan per bugje een defectticket te genereren. En daarenboven, mails leiden vooral af. Men is iedere keer geneigd naar de mailbox te kijken als er een nieuw berichtje binnenkomt.

## 8

's Middags gaat Kenneth af en toe lunchen met zijn oude vriend van zijn vorig bedrijf, David. David is een echte senior. Hij had er al 20 jaar ervaring opzitten en had Kenneth ooit opgeleid als consultant. “Je moet onthouden”, zei David, “dat testen een noodzakelijk kwaad is. Testen brengt geen geld op. Het kost alleen maar geld. Hierdoor zal men nooit grote investeringen maken in het testen. Testers krijgen nooit nieuwe bureau's, stoelen of pc's. De testomgeving is altijd de vorige ontwikkelingsomgeving. Altijd tweedehands. Zelfs niet de beste testers.”

“Hoewel dat vaak beperkend is voor je werk – je kan immers bepaalde zaken moeilijk simuleren met minder krachtige computers – moet je weten dat je tweedehands-uitstraling ook zijn voordelen heeft. Zo stoot je de programmeurs niet zo snel voor de borst. Met hun superieure infrastructuur ogen ze ‘belangrijker’ dan testers. Als ze dan feedback van je krijgen, komt dit iets minder hard aan”, aldus David.

Of die infrastructuur daar nu bij helpt of niet, er zit wel wat waarheid in David's opmerking. Kenneth had ook al ondervonden dat een low-profile-benadering de communicatie bevorderde. Hij herinnerde zich nog het vinden van zijn eerste fouten toen hij nog consultant was. Koud was de douche die hij kreeg van de ontwikkelaars toen hij hen de buglijst gaf. En hij dacht nochtans dat hij perfect had gedaan wat ze hadden gevraagd. Kijken of er nog fouten waren.

## 9

Hier viel die spanning eigenlijk best mee. Testers en programmeurs schoten redelijk goed op met elkaar. Enkelingen waren zelfs heuse vrienden. Maar het was ontstellend om vast te stellen dat die kameraadschappelijkheid wegsmolt als sneeuw voor de zon in meetings waar de teamleaders bij waren. Vooral de teamleader van de programmeurs was een genadeloze zon.

Kenneth had met een programmeur enkele fouten besproken, maar omdat de impact toch groter was dan men had vermoed, had men de teamleader gevraagd om eens samen te zitten. Toen Kenneth de fouten voorstelde reageerde de betrokken ontwikkelaar nu geheel omgekeerd. De fouten werden geminimaliseerd en zelfs in twijfel getrokken. De teamleader vond ook dat Kenneth teveel dramatiseerde. En daarenboven, waarom kwam Kenneth nu met die problemen af. Die fouten hadden al lang moeten gevonden worden. Dit stuk software was immers al eerder getest. Kenneth had gewoon bij wijze van inwerking een stuk geteste software nog eens onder handen genomen en dus fouten gevonden die de tester voor hem niet had gevonden. Het was software die Daan had getest. Kenneth voelde zich verraden, door de ontwikkelaar, en misschien ook een beetje door Daan.

Lieve zou hem die avond nog wat bijkomende begeleiding geven, als het eiland leeg was en de computers vrij.

## 10

Het was zes uur. Op de ring was de dagelijkse grote uittocht weer begonnen. Auto's bumper aan bumper, zover je kon kijken.

Kenneth zat aan zijn tweedehands-pc. Twee derde van de collega's was al naar huis. Een rust was neergedaald over de werkeilanden. Je hoorde het gezoem van pc's die niet worden afgezet. Hier en daar hing er een briefje op: "Niet afzetten".

Lieve kwam net terug van een vergadering. Moe. Het was weer een vermoeiend gesprek met haar collega, het Hoofd, hun gezamenlijke baas en enkele andere teamleaders. Het Hoofd was dan wel haar gelijke, maar toch stond hij psychologisch hoger op de ladder. Hij werkte allang bij het bedrijf en had ooit een moeilijk project gedaan samen met hun baas. Er was dus wat bonding tussen hen.

"Zitten we nog even samen?", vroeg Lieve. Kenneth keek op. "Ja, graag." Ze gingen samen een koffie halen en zochten een leeg vergaderlokaal op. Ze waren allemaal vrij op dit uur van de dag. Maar de ene was wat meer opgeruimd dan de andere.

Kenneth vertelde van zijn ervaring met de teamleader en hoe de programmeur zijn bakens met het getij had verzet. Er zat wat frustratie in zijn stem, zelfs ongenoegen, maar nog niet teveel. Kenneth wist dat hij tekort in het project zat om nu al misnoegd te mogen zijn.

Lieve luisterde aandachtig. Haar ogen straalden, zelf vermoeid, rust en begrip uit. Haar lange haar, opgestoken, met potloden op haar hoofd. Een beetje speels, voor haar. Alsof het een anachronisme was. Zwart haar, onderlijnd met zilver.

Op haar beurt vertelde ze over haar meeting. Haar verhaal klonk als dat van Kenneth. “Dynamisch gedrag” noemde ze het verschijnsel waarbij het Hoofd ook van houding wisselde. Het verraste haar al niet meer, en ze hield er rekening mee voor de meeting begon. Ze had nog geen sluitende oplossing gevonden voor het probleem maar ze kon wel al vermijden dat er foutieve besluiten werden getrokken. Ze zei dat ze het dynamisch gedrag altijd eerst liet uitdeinen zoals een veer die tot rust moet komen. Ze probeerde nooit een besluit te forceren in de meeting, maar gaf gewoon de mogelijke uitkomsten als opties mee, met haar kijk erbij. Ze stelde steeds voor om “er even over na te denken”. Meestal konden ze daarna via e-mail tot een besluit komen. En vaak kwam men terug op een eerder ingenomen positie.

Het werd zeven uur. Lieve’s stem werd wat schor. Tijd om af te ronden. Ze stonden op. Kenneth rook Lieve’s parfum. Met een kleine blos draaide hij zijn blik weg.

De file was nog niet verminderd.

## 11

Toen Kenneth de volgende morgen binnenkwam, vond hij een overstuur team. De Baas van het Hoofd en Lieve was die dag vroeger op kantoor gekomen. Toen Lieve toekwam, had hij haar meteen bij hem geroepen.

Op het matje. Want zo pakte hij haar aan. Er werd helemaal niet professioneel getest. Wat getokkel op klavieren, wat automatische testscripts, die trouwen bij elke release moesten aangepast worden. Het trok op niets. Er zaten nog steeds veel bugs in de software. De gebruikers waren niet tevreden. Het betoog van de Baas was als een dikke mousselinesaus die niet wil binden.

De Baas had de vorige dag een telefoontje gekregen van zijn baas, de CEO. Een belangrijke klant had net een marketing-mailing uitgevoerd met behulp van hun contactdatabase en nu bleek dat zeventig procent van alle verstuurd brieven, zo'n vijfduizend stuks, naar de correspondenten was verstuurd die deze boodschap net niet hadden moeten ontvangen. Een kleine ramp dus. Een grote eigenlijk, voor de klant.

Shit always comes down, dus had de CEO gebeld, de Baas had het Hoofd erbij gehaald, want deze was uiteindelijk de maker van het product. En het Hoofd had gezegd dat het testteam had gemeld dat alles ok was. Hij zei er ook bij dat hij dat weliswaar betwijfelde maar dat hij uiteindelijk de verantwoordelijkheid van het testteam niet mocht afnemen. Hij had dus de software gereleased.

Shit always comes down. En het is zoals de stokken in het kippenhok. Op de onderste stokken ligt altijd de meeste shit. Zo ook hier. Lieve zat op de onderste stok.

Daarenboven had de Baas net zijn budget gekregen voor het volgende jaar. Zijn budget werd met dertig procent verminderd. Het Sigma-projectteam moest kleiner worden. Om te beginnen moesten de “consultants”, de externen dus, eruit. Zo snel mogelijk.

En dat testen, dat was ook een probleem. Dat kostte veel geld en leverde geen toegevoegde waarde. Want de klanten waren nog altijd ontevreden. Mocht er niet getest worden, dan was de klant toch ook ontevreden.

De boodschap aan Lieve was dus dubbel: “je doet je werk slecht. Corrigeer dit en weet dat je testbudget met dertig procent zal dalen”. De managementlogica zegt immers dat als je overal dertig procent reduceert, je totale organisatie met dertig procent gereduceerd wordt.

Zelfs voor Lieve was dit even teveel. Ze had meteen haar team samen geroepen en het probleem voorgelegd. Ze hield er immers een open en participatief management op na.

Net toen Lieve haar uitleg aan het geven was, was Kenneth toegekomen.

Participatief management...een probleem in dergelijke situaties. Want het team, waarvan nu extra inspanningen van verwacht werd, was nu op de hoogte dat er een inkrimping van dertig procent zou komen. Niet bepaald motiverend.

Maar iedereen was geëngageerd. Verbouwereerd weliswaar. Onthutst. Gefrustreerd, zelfs kwaad, maar geëngageerd. Niemand was extern. Als er dus dertig procent daling moest worden gerealiseerd, dan zou het dus over ontslag gaan.

## 12

Kenneth stond aan de koffie te mijmeren over de meeting. Hij was laatst bijgekomen in het team. Als eerste zou hij natuurlijk terug moeten vertrekken. En hij kwam net van een afdeling waar hij overtuigd was. Hij zou wellicht geen andere plaats in de organisatie meer kunnen krijgen. En het gaat niet zo goed op de arbeidsmarkt.

Zijn gepeins werd onderbroken door Miriam. Miriam stond wat verderop in de cafetaria te praten met een ontwikkelaar. Van alle testers had Miriam de meest nauwe relatie met de ontwikkelaars. Maar nu was het wel heel nauw. Iedere mens schijnt een soort virtuele persoonlijke bubble rond zich te hebben die zo'n 60cm in straal bedraagt. Met andere woorden, als je dichterbij iemand gaat staan, of er komt iemand dichterbij jou staan, dan voel je je ongemakkelijk. Aangetaast in je privé als het ware.

De bubble van Miriam was 10cm. Maar de bubble was eigenlijk weg, zoals met een zeepbel. Als je er door prikt dan is hij weg. De ontwikkelaar had geprikt, want hij had, schijnbaar onschuldig, iets aangeduid op een document dat ze samen aan het bekijken waren. Daarmee had zijn rechterarm langs de linkerborst van Miriam geschuurd. En toen hij met zijn vingers op de tekst wees, bleef de borst steunen op de arm. De bubble was uiteengespat. En zo ook het gemijmer van Kenneth.

Ze droeg haar zwarte rokje. Het "probleemrokje". Meestal, als er wat problemen waren tussen het testteam en de ontwikkelaar, kwam Miriam in haar zwarte rokje werken.

Blijkbaar was dit de geschikte werkkledij om problemen op te lossen, want dat werkte altijd.

Een zwart rokje. Met een zeer mooi ruw geweven bruin/zwart/grijs gilé erboven. Door de ruw geweven stof zag je haar zwart satijnen bh.

Miriam merkte Kenneth op. Schijnbaar onverstord maakte haar borst zich los van de arm van de ontwikkelaar. Toch enige vorm van onwennigheid. Ze acteerde niet naturel. “Een goed acteur zorgt ervoor dat hij niet speelt”, zei wijlen Ann Petersen.

Ze kwam op Kenneth af. Beetje overleggen over de crisis van het testteam. Daarbij niet verwijzend naar haar gesprek met de ontwikkelaar. En ze kwam ook in de 60cm zone van Kenneth. Niet omdat haar borsten steun nodig hadden. Maar een poging om haar intiem gedrag van daarnet wat te neutraliseren. Immers, als ze dat ook met Kenneth doet, dan lijkt het voor hem “normaler”.

## 13

Aan zijn bureau overwoog Kenneth zijn opties. Moest hij nu toch gaan solliciteren? Hij had op dat punt gestaan net voor hij werd toegewezen aan Sigma. En omdat hij niet echt van risico's hield, was hij opgelucht geweest toen hij hier naartoe mocht komen. Wat nu?

Lieve kwam op hem af. Ze vroeg hem even mee te komen. "Het is al zover", dacht Kenneth, "ik mag vertrekken."

In tegenstelling tot zijn verwachting vroeg Lieve hem om hulp. Hij was nieuw, had Lieve gezegd, en daar wou ze gebruik van maken. Hij kon nog zonder vooroordeel en vooral zonder emotionele binding het project analyseren. Ze vroeg hem of hij de activiteiten van het testteam kon analyseren. Hij was even verbouwereerd.

Kenneth sprak af met David om te lunchen. "Lieve heeft dat goed bekeken", zei David. "Ik zou om te beginnen gewoon eens een lijstje van de zaken die me verrasten toen ik in het project zou binnengestapt zijn. Ik heb al ondervonden dat dit meestal ontspoorde initiatieven zijn die ooit opgezet werden om bepaalde problemen van toen op te lossen. Altijd goed bedoeld, soms zelf even doeltreffend, maar vaak symptoombestrijdende acties. Die trouwens met veel inspanning werden toegepast zodat nu niemand ze nog durft in te twijfel te trekken. En wie een tijdje in het project zit, ziet deze absurditeiten niet meer.

En daarnaast, ging David verder, "waarom? Als men kostenreducties voorstelt, waarom? Ik moet altijd aan mijn leraar denken op school die zei dat de makkelijkste en meest

doeltreffende manier om kosten te reduceren, de fabriek sluiten is. Maar dan stopt de productie ook natuurlijk. Veel kostenreducerende maatregelen hebben een productieverminderend effect. De vraag is altijd of men zich dat echt bewust is?"

Gemakkelijker gezegd dan gedaan. Vooral dat laatste. Je stelt niet zomaar een beslissing van hogerhand opnieuw in vraag. Zeker niet als ze zo dringend met zoveel overtuiging werd gebracht.

## 14

Aan zijn bureau begon Kenneth met zijn lijstje bedenkingen.

Daan zat voor hem, zoals gewoonlijk gezapig te testen. Het leek alsof Daan nooit onder druk stond. Daan had het precies altijd onder controle. En hij stond nooit in het centrum van één discussie.

Dat was misschien zijn eerste bedenking: hoe komt het dat Daan, als enige in het testteam, geen last ondervindt van de kritiek die de testers krijgen.

Hij testte, vond af en toe een fout, maar slaagde erin om steeds zijn werk binnen de verwachte tijd af te maken, zodat releases nooit door zijn toedoen in het gedrang kwamen.

Daan test niet goed: te gemakkelijk, waardoor hij geliefd is bij de programmeurs.

...

Wat is zijn tweede bedenking?

## 15

Ondertussen ging het niet zo goed met het project. Er moest deze week een release buiten, maar die zou nooit op tijd klaar zijn. Zeker niet als de testers er nog eens moesten doorgaan. Ze hadden de release weliswaar getest, maar men de gecorrigeerde versie was nog niet hertest. Er waren al stemmen opgegaan om die hertest te schrappen, nu en ook later. Immers, er zou toch geen tijd zijn om eventuele aanpassingen te doen. Dus waarom dan nog testen?

Bovendien veroorzaakte het testen meestal nog meer ellende. Vorig jaar was het zelfs zo erg dat er gemiddeld drie nieuwe fouten werden geïntroduceerd bij elke fout die hersteld werd. Drie! Da's een heel pak. Hoe meer er dus getest werd – en hersteld- hoe slechter de kwaliteit.

De deadline was vrijdag. Het was nu maandag en de regressietest moest nog beginnen. Het installeren van de release en het stabiel brengen van de werkomgeving duurt, in het beste geval, twee dagen. Drie dagen bleven dus over om een regressietest uit te voeren. Een test die, als hij vlot loopt, zeven dagen nodig heeft.

Het Hoofd was zich al komen moeien en had al geopperd om meerdere testers parallel aan het werk ten zetten. Hij was bij Kurt en Miriam geweest. Het had Lieve geërgerd. Het Hoofd had zich niet met haar team te bemoeien. Hij zou beter naar zijn eigen planning kijken.

Het managementadvies was weer even eenvoudig als verraderlijk. Meerdere testers inzetten veronderstelt dat meerdere mensen voldoende voorkennis hebben om dat werk

te doen. Daarenboven stijgt de complexiteit om al dat testwerk te managen kwadratisch met de toename aan testers. Testen, die op dezelfde systemen worden uitgevoerd, interfereren gemakkelijk met elkaar. Fouten kunnen makkelijk dubbel gerapporteerd worden... Iedereen moet goed weten hoe hij een fout rapporteert...

Niet alleen deze release moest buiten. Er was bovendien ook een upgrade gepland van alle Franse klanten. Die hadden een speciale versie die min of meer parallel met de standaardversie werd ontwikkeld. Eén van de eerste Franse klanten, een winkelketen of zoiets, die een heel groot order had geplaatst, had indertijd een pak eigen functionaliteiten geëist. Gevolg was dat heel de Franse markt nu met die aangepaste versie werkte en dat men bij Beetleware dubbele maintenancekosten had. Daarenboven kreeg die grote klant, de eerste drie jaar gratis onderhoud. Geld werd in Frankrijk niet verdient. Het stond echter goed op de referentielijst van het bedrijf. Beetleware was internationaal actief.

## 16

Er had een geheime meeting plaats gevonden. Het Hoofd had zijn teamleaders rond zich verzameld. De planning was onhaalbaar met dat testteam ertussen. Als ze het testteam voor deze release ertussenuit haalden, dan konden ze twee vliegen in één klap slaan. De software zou niet meer slechter worden, want er zouden geen nieuwe herstellingen meer gemaakt worden en ze zouden de uitgespaarde testtijd kunnen gebruiken om meer te programmeren.

De teamleaders hadden sowieso al gezegd dat deze versie beter was dan alle vroegere releases. De risico's waren dus beperkt. Als het lukte, zouden ze een schitterende prestatie neergezet hebben. Die testers zouden eens echt duidelijk gemaakt worden wie hier de echte IT-professionals waren. Die testers... de helft had niet eens ooit echt geprogrammeerd.

De teamleaders stemden in. Alleen moesten ze er nu wel voor zorgen dat hun programmeurs niet uit gewoonte met de testers gingen praten. Dat kon immers wel eens uitlekken. De programmeurs zouden niet ingelicht worden over het feit dat er niet zou getest worden. Als ze niets wisten dan konden ze ook niets verklappen: de vergaderzaal rook naar samenzwering.

## 17

Lieve had een dochter, een hond en was gescheiden. Haar verleden had wel wat littekens nagelaten, maar het waren er van het soort die je mooier maken. En geen nieuwe relatie. Wat menig collegae hoop gaf. Een paar hadden in het verleden, schoorvoetend of heel expliciet, wat avances gemaakt. Maar Lieve was er nooit op ingegaan. Wat de begeerte alleen maar aanwakkerde natuurlijk. “Geen collega’s” was haar devies. Maar eigenlijk was er niemand die haar interesseerde.

Voor Kenneth deed ze daarentegen wel een kleine inspanning. Sedert Kenneth op het project zat, stond ze ’s morgens wat langer voor de kleerkast. Ze was plots, onwillekeurig, bewuster bezig met haar verschijning.

“Aanstellen”, had ze zelf altijd gezegd. Relaties beginnen altijd met een “Hollywood”-fase. Of het nu het begin is van een privé-relatie of een werknemer/werkgeversrelatie, in het begin stellen beide partijen zich wat aan. Wordt de relatie goed, dan vervangt vertrouwen en respect de oppervlakkigheid en per definitie het oneigenlijk contact. Zoniet, dan wordt de oppervlakkigheid alleen maar vlakker, en dus meer glad. Uiteindelijk heb je geen houvast meer en ga je op je bek. En tussenin komt er misschien wat sex.

Nu deed ze het zelf ook, aanstellen. Klein, niet met grootse plannen of verlangens, maar ook niet onschuldig.

## 18

Kenneth had zijn eerste analyse klaar. Hij ging met Lieve punt voor punt door zijn rapport. Hij had zijn verslag “De ongeschreven regels van Sigma” genoemd.

Ze zaten samen in een lege vergaderzaal. Lieve had het rapport al via mail ontvangen en diagonaal doorgenomen. Daarbij was ze tegelijk verrast -onthutst eigenlijk- en tevreden. Tevreden omdat Kenneth een sterk en helder rapport had voorgelegd.

Zijn besluit had hij samengevat als: “De testers hebben geen enkele toegevoegde waarde en lijken uitsluitend als versiering van het project”.

Zijn eerste motivatie van dit besluit was gebaseerd op het feit dat er slechts 5 testers waren op een project van meer dan 20 programmeurs, analisten buiten beschouwing gelaten. Samen testten ze ongeveer 60% van de schermen en maar 30% van alle codes. De testers vonden ongeveer 1,1 fouten per uur, wat overeenkwam met het industriegemiddelde van softwaretesten. 1,1 fout per uur betekent vooral dat er gewoon teveel fouten in het product zaten ten opzichte van de beschikbare tijd om ze te vinden. En met een verhouding van 5 op 20 was een gevonden fout slechts een druppel op een hete plaat. De kwaliteit werd niet merkbaar beïnvloed door de gevonden – en herstelde – fouten.

Ten tweede had hij vastgesteld dat de testers aan het project waren toegevoegd omdat de “methodologie” het voorschreef, niet omdat men erin geloofde. Voorheen bouwde Beetleware software volgens een eigen recept. Daarin was

testen een activiteit die deels door de programmeurs en deels door de mensen van support werd uitgevoerd. Een audit van een consultancy bedrijf had ervoor gezorgd dat heel het recept om software te maken, formeler was geworden. Testen hadden daarbij een nieuwe plaats gekregen, met aparte personen. Maar die voorgestelde methodologie was zoiets als een recept uit een kookboek. Een professionele kok gebruikt ze niet en wie ze letterlijk toepast, merkt steevast dat de eetlepel die vermeldt staat, niet overeenstemt met het bestek dat je in huis hebt.

Verder bestond Kenneths rapport uit een pak regels die blijkbaar door iedereen werden nageleefd, maar die niet op papier stonden. Zo bleek dat tegen het Hoofd nooit geargumenteed werd. Iedereen die ooit het Hoofd voor het hoofd had gestoten (Kenneth vond dit een leuke vondst) was op één of andere manier uit de onderneming verdwenen.

Teamleaders stonden boven de wet. Ze volgden de methodologie niet en werden er niet op aangesproken. Ze leverden zelden volgens afspraak de verwachte resultaten. Ze motiveerden ook nooit waarom een afspraak niet werd nagekomen. Het leek alsof die afspraak niet bestond.

## 19

Lieve hoorde met grote ogen het verhaal aan. Alles wat ze vernam, klonk zeer vertrouwd in de oren. Maar toch was het confronterend en stuitend. Ze was deel van het management en ze had het uiteindelijk ook allemaal laten gebeuren.

Een mengeling van bewondering en schaamte. Ze had het moeilijk de juiste houding aan te nemen ten opzichte van Kenneth. Ze stond aan het hoofd van de “versiering” van het project. Niet meteen een indrukwekkende positie. Kenneth had het allemaal heel raak geanalyseerd. Zou hij ook over haar een analyse gemaakt hebben, die niet in het rapport stond?

Ze overliepen alle opmerkingen. Hier en daar argumenteerde Lieve. En steeds had ze het gevoel dat ze de situatie “verdedigde”. Kenneths opmerkingen waren ontegensprekelijk terecht.

## 20

Die avond werd er door een teamleader, samen met twee ontwikkelaars, een release gemaakt. Zonder testen. Een illegale versie. Morgen zou die naar de bètaklanten vertrekken.

Bètaklanten: die klanten die zich bereid hadden verklaard om als eersten de nieuwe software te ontvangen tegen de voorwaarde dat ze weliswaar met de bugs moesten leven. Tot veel van die klanten behoorden trouwens de grote accounts.

De illegale cd lag klaar op het bureau van de teamleader.

Kenneth stond op het punt naar huis te vertrekken. Hij had een rapport klaar voor de ontwikkelaars en liep nog snel het bureau van de teamleader binnen. Hij legde het rapport in het IN-bakje. Er lag een cd in. Release v5.3.11b. Aanvankelijk besteedde hij er geen aandacht aan, maar toen hij het lokaal verliet, begon het door te sijpelen. Hij had net een rapport opgeleverd over versie v5.3.8. Wat deed die versie v5.3.11b daar dan?

Hij keerde op zijn stappen terug en ging de open space van de ontwikkelaars binnen. Een paar programmeurs waren nog aan het werk. Hij vroeg hen naar de versie waarop ze aan het werken waren? 5.3.12, sedert vandaag. Deze ochtend was immers v5.3.11 afgewerkt.

Ze wisten niets van de status van die versie. Ze wisten niet of de versie al dan niet getest was. Voor zover zij wisten

was deze versie zoals de andere: geprogrammeerd, getest, ingeblikt.

Kenneth ging terug naar het testlokaal. Lieve was nog aan het werk. Hij liep op haar af en vertelde haar over de cd. Lieve was verbaasd. Ze stond op en samen gingen ze naar het kantoor van de teamleader. De programmeurs waren ondertussen vertrokken. De cd lag nog steeds in het IN-bakje.

Lieve nam de cd mee. Morgen zou ze de teamleader erover aanspreken. Ze voelde nattigheid.

Dit was het uitgelezen moment voor Lieve. Waarom Kenneth niet vragen om mee uit te eten? Het was al na zevenen.

Hoewel hij even schrok van de vraag, stemde hij meteen in. Ze gingen samen uit eten.

## 21

Ze zaten in een beetje oubollig Italiaans restaurant. Het was eigenlijk de tijd van de “trendy lounge bars”. Bruine, of donker getinte resto’s, met vaak zachtgloeiend rood of blauw licht, strakke interieurlijnen en lounge muziek.

Lieve verkoos de “Firenze”. Een Italiaans restaurant, zo oud als de straat. Met klaar en gelig wit licht dat uit oude lampkapjes komt. Maar het eten is er goed en vooral, de service is zo tof. De kelners zijn enorm charmant, ze nemen nog heel beleefd je jas aan, je krijgt een drankje bij de rekening, die trouwens maar een fractie meer is dan bij een studentenresto. En je zit altijd “apart”. Alle tafels zijn deftig van elkaar gescheiden, meestal met een gemetst muurtje, met weliswaar plastieken planten erop.

De Calzone is er subliem. Eén Calzone en een Cannelone. Als dessert een sabayon met ijs, aan de tafel gemaakt. Een hele belevenis. Telkens als er een klant een sabayon bestelt, komt er een oude dikke kok uit de keuken. Men zet een plaatje op met een Italiaans nummertje over een sabayon en Marcello begint te roeren. Heel het restaurant weet wanneer je een sabayon bestelt. Gemetselde muurtjes ten spijt. Koffie met de obligate amaretto.

## 22

Ze hadden het grootste deel van de avond niet over het werk gepraat. Kenneth vertelde zo een beetje hoe hij bij Beetleware was terecht gekomen, zijn reisverhalen van vroeger en waarom hij uiteindelijk van zijn vorige job in de firma weg moest. Lieve had er inderdaad al van gehoord dat de supportafdeling waarvan hij deel uitmaakte en waar hij testte, ingekrompen was.

Hij vertelde over zijn hobbies. Vroeger veel elektronica en zo. Schakelingen maken. Allerlei dingen voor zijn tweede hobby, een discobar runnen. Van zijn achttiende tot enkele jaren geleden was hij “Dj-K”, iemand die allerlei parties opluisterde met zijn muziek. Nu houdt hij zich vooral bezig met zijn pc. Na kantooruren schrijft hij hier en daar een programmaatje voor verenigingen en zo.

Lieve praatte over haar carrière bij Beetleware, die nu 4 jaar liep. Maar ze zei niet zoveel, ze wou vooral Kenneth leren kennen.

Het gesprek nam een werkwending. Ze praatten over het rapport en de programmeermiserie. Wat ook de besluiten van zijn mooie rapport zijn, feit is dat er heel veel fouten in de software zitten, dat ontwikkelen chaotisch verloopt en dat de relatie met de testers helemaal niet goed is. Er heerst een echte vijandigheid. Kenneth ging op zijn vaststelling in. De programmeurs, en zeker de teamleaders, zijn eigenlijk slecht.

“Allez, ze zijn toch echt niet professioneel”, zei Kenneth.

## 23

“Het is maar de vuilnisbak”, zei Lieve.

“Hoe bedoel je?”, vroeg Kenneth, zich afvragend of hij nu iets gemist, dan wel verkeerd gehoord had.

“Het is maar de vuilnisbak. Ik heb ooit eens een hele rare ontmoeting gehad”, verklaarde Lieve.

“Ik had op een keer een halve dag verlof genomen met mijn vriendje. Het was op mijn eerste werk, ik was een paar maanden afgestudeerd. Het was de eerste heel mooie lentedag van het jaar, ergens half maart of zo. Het was kalm op het werk. Economische recessie, zoals nu een beetje. Ik wou naar de zee. Mijn vriendje zat in het laatste jaar van zijn studies. Een halve dag brossen met mij zag hij wel zitten.

Wij naar de kust. Wandelen, wat onnozel doen, je kent dat. Het was midden in de week. Er was geen kat op het strand. Nog een paar koppeltjes zoals wij, wat bejaarde wandelaars en één motard. Een motorrijder, helemaal in het zwart gekleed, in het leder. Zwarte zonnebril. We waren hem al tegengekomen. Koen, mijn vriend, wou mij wat plagen of zoiets, doen alsof hij me in de zee zou gooien. De motard passeerde ons en glimlachte.

Wat later lagen we op een golfbreker en de motard kwam terug. Hij was blijkbaar ergens naartoe gewandeld en keerde nu langs het strand terug. Hij zag ons liggen, ik met mijn hoofd in Koen's schoot, wat lachen en praten. Het was een heel toffe dag. En de motard moet dat gezien hebben. Want hij kwam naar ons toe en sprak ons aan. Wij natuurlijk

heel verrast. Hij vroeg ons of we ons amuseerden. We waren een beetje verbouwereerd, maar zeiden ja. We konden precies niet anders dan hem antwoorden. Hij imponeerde ons wel.

“Wel”, zei hij, “onthou vandaag. Prent deze dag goed in je geheugen, zodat je hem nooit vergeet. Want op een dag zal je de herinnering nodig hebben. Hij – hij wees naar Koen – zal op een dag de vuilnisbakken niet buiten gezet hebben. En jij – ik dus – zal kwaad worden, hem uitschelden en zeggen dat hij een luiarik is, die nooit in het huishouden helpt.”

“Wel”, ging hij verder, “Voordat je hem een luiarik noemt, denk aan vandaag, aan nu, het moment dat je hem blijkbaar ongelofelijk goed vindt. Aan dit mooie weer en alles hier op het strand. En vraag je dan af of hij nu echt lui geworden is, na al die jaren, of dat hij gewoon de vuilnisbakken vergeten is. En zeg dan wat je wilt zeggen. Weet, als je hem buiten gooit, dan zal een ander waarschijnlijk ook de vuilnisbakken eens vergeten.”

We keken heel verbaasd. “Waarom?” was zowat het eerste en enige dat ik kon uitbrengen. “Ik was het vergeten,” zei hij, “en heb gezegd dat ze lui was. En nu moet ik alleen de vuilnisbakken buiten zetten.” En hij ging weg. Even onverwacht als hij gekomen was. We bleven hem verstomd nakijken. Het was een heel rare situatie. We hebben nog heel de namiddag over vuilnisbakken gepraat. We geloofden die dag nog méér in elkaar.”

“Wel, die vuilnisbakken, dat zijn de fouten die de programmeurs maken. Ze doen dat niet met opzet. Ze doen hun best, maar ze zijn niet perfect. En we weten dat eigenlijk

wel, jij ook, maar we vergeten het zo snel, zeker als het slecht gaat. Maar omdat die motard mij toen zo verraste op het strand, vergeet ik dat nooit meer en maak ik altijd eerst de bedenking wat het probleem echt is, voor ik iemand met de vinger wijs. Straf hé? Vuilnisbakken.”

Kenneth keek Lieve geboeid aan.

“Waar is Koen nu?”, was het eerste hij kon uitbrengen.

“Oh”, lachte Lieve, “Koen en ik zijn na zijn examens uit elkaar gegaan. We waren uiteindelijk toch niet voor elkaar gemaakt. Ik denk dat hij nu ergens in Antwerpen de vuilnisbakken buiten zet”.

## 24

Kenneth ziet vanop zijn bureau Lieve grote armbewegingen maken in het kantoor van de teamleader. De CD in de hand. Hij voelt er zich ongemakkelijk bij. Hij heeft het vuur aan de lont gestoken. Hij heeft de CD gevonden en Lieve ingelicht. Wellicht zal dat wel op tafel komen. En dan kon hij het wel schudden om nog overeen te komen met de programmeurs.

Miriam zit naast hem, ook te kijken naar het vertoon achter het glas. Melkglas, tot zowat ter hoogte van de schouders, waardoor alleen rondzwaaiende armen echt zichtbaar zijn. Lieve zelf is een wazige vlek die overvloeit in de vlek van de teamleader.

De deur gaat open. Lieve stapt woedend buiten, de gang in. De teamleader zit neer.

## 25

Ze loopt Kenneth en Miriam voorbij. De gang door. Het kantoor van haar baas binnen. Even later gaat ook de Hoofd het kantoor van de Baas binnen. Een uur gaat voorbij.

De geheime afspraak, de illegale release is inderdaad aan het licht gekomen. Ook de Baas is, tot Lieve's verbazing, woedend.

Desalnietemin neemt hij een bijzondere beslissing. De release mag uiteraard niet vertrekken. Het Hoofd moet deze namiddag met de verkoopsdirecteur samenzitten om een communicatie naar de Bèta-klienten en de Franse klienten voor te bereiden. Sales moet immers uitleggen waarom de verwachte releases niet zullen komen. Er komen nieuwe release-data. De Baas zal het Hoofd coachen in dit gesprek. Lieve moet ondertussen haar planning herbekijken in functie van de nieuwe deadlines. Morgenochtend zullen het Hoofd, de teamleaders en Lieve een nieuwe algemene planning maken. Samen. De Baas verwacht tegen morgenmiddag een planning door Lieve en het Hoofd samen voorgesteld.

Na een uur staan het Hoofd en Lieve weer buiten. Ze spreken mekaar niet aan, gaan elk hun weg.

## 26

Lieve zit over haar bureau gebogen, handen letterlijk in het haar. Voor haar een pak spreadsheets en enkele vellen Gant-charts. Een onmogelijke planning, die nu binnen de 24 uur moet herzien worden, in samenspraak met die dikkop van een Hoofd. Ze ziet geen uitweg.

Daan zit rustig te testen. MP3-speler op het hoofd. Kurt zit scripts te schrijven. Miriam is een testdesign aan het maken en Kenneth is de buglijsten aan het overlopen. Iedereen heeft zijn werk. Het eiland zoemt.

Kenneth had David op een instant messegner gecontacteerd. Om te lunchen. Hij brengt hem kort op de hoogte van de crisis. Kenneth stelt Lieve voor te gaan lunchen met hem en David. Een beetje tegen haar zin, gaat Lieve mee. Liefst zou ze gewoon aan haar bureau willen blijven zitten en een oplossing zoeken. Maar ze voelt dat die er niet zomaar zal komen.

## 27

Twee dagschotels zonder soep. En een Ceasar's salad. Voor Lieve. David luistert aandachtig naar het relaas van Lieve. Ze vertelt het niet omdat ze denkt dat David haar een oplossing zal geven. Ze wil het gewoon eens uit haar lijf krijgen.

Ze heeft geen idee hoe ze een nieuwe planning kan ophoesten. Het huidige werk is goed georganiseerd. De latere deadlines maken zelfs dat haar planning alleen maar realistischer wordt. Ze heeft dus geen nood om met het Hoofd samen te zitten en alles terug overhoop te gooien. Maar ze moet met hem samen zitten. Opdracht van de Baas.

“Er zijn dus heel wat fouten die nog opgelost moeten worden, als ik het zo hoor?”, vraagt David aan Lieve.

“Absoluut. We hebben nog 500 openstaande defecten, waarvan er bijna geen één door development van dichtbij bekeken is. En wie weet wat we niet gaan vinden in die laatste cd die we ontdekt hebben.”

“Wat is het doel van je werk?” gaat David verder.

Lieve vraagt zich af of de vraag retorisch is. Maar David stopt en blijkt echt op een antwoord te wachten.

“Fouten vinden”, zegt ze, “zodat, als ze hersteld zijn, de kwaliteit van de software verbetert”. Ze vindt het een banaal antwoord op een banale vraag.

“Fouten vinden is inderdaad wat de testers moeten doen”, bevestigt David, “en het doel is inderdaad het

realiseren van een betere kwaliteit. Is het zo dat het vinden van meer fouten, de kwaliteit zal verhogen? Of dat de kans dat betere kwaliteit wordt gerealiseerd tegen de nieuwe deadline, zal verhogen?”

“Zolang de fouten niet hersteld worden, zal de kwaliteit niet verhogen” brengt Kenneth in.

“Juist. Het bereiken van je doel – betere kwaliteit – hangt nu af van de mate waarmee men fouten zal herstellen. Je zit immers al met een hele pak fouten die klaar liggen om bekeken te worden.” David stopt even, herneemt, “Je honger wordt niet gestild door veel gerechten klaar te maken of veel eten te kopen, maar door wat je maakt of koopt ook daadwerkelijk op te eten. Als je grote honger hebt, dan heb je neiging om veel eten te kopen. En altijd merk je dat je niet alles op krijgt. Beter minder te kopen of koken en meteen beginnen eten.

Maar nu moet ik dringend naar mijn volgende afspraak” sluit David af, “het was aangenaam kennismaken.” Hij geeft Lieve een hand en knikt even naar Kenneth. “Ik betaal wel voor ons drieën. Drinken jullie nog maar een koffie”.

Kenneth en Lieve blijven alleen achter.

## 28

“Eten als je honger hebt, wat wil hij daar mee zeggen?”, denkt Lieve half luidop.

“Fouten herstellen”, redeneert Kenneth mee, “de fouten die we nu al kennen eerst herstellen vooraleer we nog meer naar nieuwe fouten gaan zoeken.

David zegt dat we eigenlijk moeten stoppen met testen, zoals stoppen met ons karretje vol eten te laden. We moeten herstellen, allez, de ontwikkelaars moeten eerst herstellen, want we kunnen dat niet.”

“Jamaar”, sputtert Lieve, “als wij niet kunnen herstellen, dan heeft het toch geen zin dat wij stoppen met testen? Dat heeft toch geen impact op de herstelling...”

Het is stil. Kenneth weet het ook niet. Lieve heeft wellicht gelijk. De redenering van David gaat slechts op als je zelf winkelt en zelf eet. Maar als jij koopt of kookt, en iemand anders eet je gerecht op, dan is er geen probleem om je kar vol te laden...

“Tenzij”, begin Lieve opnieuw, “ we natuurlijk met de vrijgekomen tijd ander werk doen. Kurt bijvoorbeeld, die was aangeworven om te programmeren en werd onvrijwillig aan ons toebedeeld. Als we hem bij de programmeurs zetten, dan kunnen ze sneller herstellen.”

Kenneth luistert. Zo had hij het nog niet bekeken.

Plotseling lijkt Lieve tien jaar jonger. Ze is opgewekt, opgewonden eigenlijk. Alsof ze het plots het licht heeft gezien.

Ze lacht.

“Kom”, zegt ze, “ik weet wat we moeten doen. Laat ons met het testteam samenzitten.”

Ze is mooi.

## 29

Lieve heeft haar team een radicaal voorstel gemaakt. Kurt gaat naar de programmeurs om te helpen met de herstellingen. Of eerder om nieuw werk uit handen te nemen van de programmeurs zodat deze hun programma's kunnen verbeteren. Allez, het Hoofd moet maar zien hoe hij Kurt het beste inzet.

Daan wordt gereserveerd om regressietesten te maken. Hij moet scripts maken, kleintjes, en de programmeurs bijstaan door bij elke herstelling, direct een aantal testen te kunnen doen.

Miriam wordt ingezet als liaison-officer. Ze moet de analisten helpen met het interpreteren van de bugs, het helpen aanmaken en sorteren van de buglijsten. Ook met het nuanceren van de soms sterke bewoordingen in de bugbeschrijvingen. In heat of fight is men soms ongenueanceerd. Miriam geniet een grote acceptatie bij de ontwikkelaars. De meeste zijn natuurlijk mannen. Echte pubers als het op mooie vrouwen aankomt.

Kenneth moet het proces opvolgen en helpen met het plannen. Vooral de doorstroming van de bugs. Checken waar ze eventueel blijven steken, een beetje in gaten houden of de ontwikkelaars voldoende met herstellen bezig zijn en haar informatie geven.

Dat laatste zal een moeilijk punt worden. Het Hoofd voldoende overtuigen dat hij bij zijn ontwikkelaars de klemtoon op herstellen legt. Als hij blijft nieuwe requirements

ontwikkelen, dan heeft het allemaal geen zin natuurlijk. “David zou beter eens met het Hoofd praten”, denkt Lieve.

De testers begrijpen het nog niet helemaal. Wordt het team nu uit elkaar gehaald? “Ik moet toch niet voor het Hoofd gaan werken”, was al een reactie van Kurt.

Lieve legt uit dat het tijdelijk is, dat het geen formele reorganisatie is, maar we gewoon allemaal samen als één team die release proberen buiten te krijgen.

Haar collega's zijn nog niet helemaal overtuigd. Even gas terug nemen, koffietje halen. En er dan nog eens doorgaan.

Terwijl ze om haar chocochino gaat, hoort ze Kenneth haar verhaal in zijn woorden brengen.

## 30

Dinsdagochtend. Meeting tussen Lieve en het Hoofd. Beide onwennig. Lieve ondertussen met zin, ze heeft plannen. Het Hoofd niet. Hij vond zijn plannen best ok. En die nieuwe deadlines waren ook voor hem een zegen. Om verder te ontwikkelen.

En nu vertelt Lieve hem dat hij zou moeten stoppen met ontwikkelen en beginnen herstellen. Geen denken aan.

Lieve had haar voorgenomen haar vooroordelen niet mee te nemen naar de onderhandelingstafel. Ze probeerde zich vuilnisbakken voor te stellen. Een hele vuilnisbelt. Alleen, ze mocht er niet aan denken dat ze met het Hoofd lief en leed zou delen. Zo ver mocht ze de metafoor niet trekken.

Na het uiteenzetten van haar plan, geeft ze het Hoofd alle ruimte om te reageren. Ze wacht zelfs te lang, toont teveel begrip, want het Hoofd denkt dat ze niet meer luistert. Hij windt zich wat op.

Ze laat het over zich heen komen, maar geeft toch wat weerwoord.

Ze vertelt het verhaal van het eten en de honger. “Het heeft geen zin nieuwe functies te schrijven die we toch niet zullen kunnen testen en die dan samen te releasen met de functies waarvan we weten dat er nog fouten in zitten”, zegt ze.

“Maar”, legt ze uit, “je krijgt van mij een extra ontwikkelaar, Kurt, zodat je meer capaciteit hebt om te herstellen en toch wat verder te ontwikkelen”.

“Kurt, die halve programmeur. Ik moet alleen echt programmeurs hebben. Geen romanschrijvers,” Het Hoofd op zijn best.

Later herneemt hij zich wat. Een uur gaat voorbij. Tijd voor een sigaret. Zonder uitleg staat hij op en loopt hij de gang in. Lieve kijkt hem na. Hij gaat naar de parking. Pikt onderweg een teamleader op.

“De boer”, denkt ze. En dan even later, “hij neemt toch zijn teamleaders mee om te overleggen. Dat betekent dat hij het toch interessant genoeg vindt om over na te denken. Even wachten”, zucht Lieve. De motard met de vuilnisbakken helpt haar weer geduld te hebben.

## 31

Twee weken later – het project loopt nu zoals Lieve het min of meer had voorgesteld. Het Hoofd had, na overleg met zijn teamleaders, ingestemd om Lieve's voorstel te volgen. Eén van de teamleaders had het Hoofd gezegd dat Lieve eigenlijk bottleneck-theorie aan het toepassen was, waarna hij een hele uitleg had gegeven over Goldratt en zijn Doel<sup>2</sup>. Het Hoofd had ingestemd.

Heimelijk hoopte hij weliswaar dat de teamleaders toch zouden proberen hier en daar de belangrijkste nieuwe functies in te programmeren. In de praktijk gebeurde dit quasi niet. Weliswaar niet omdat de teamleaders nu zo geloofden in het belang van herstellen, maar omdat ze gewoon geen tijd hadden. Er waren teveel defecten die moesten behandeld worden.

Op zich zou deze situatie behoorlijk irritant geworden zijn voor de teamleaders en de ontwikkelaars, maar de zet om Miriam als tussenschakel te laten werken, deed zijn dienst. Ze aten allemaal, bijna letterlijk, uit haar hand. Miriam was gelukkig met al die aandacht – soms iets teveel, zeker als ze weer eens werd uitgenodigd voor een diner na het werk. Lieve en het Hoofd waren tevreden. Kortom, een goede oplossing.

---

<sup>2</sup> Zie, The Goal (Het Doel), E. Goldratt

## 32

Met Kenneth vlotte het stukken minder. Hij had van Lieve de opdracht gekregen om een soort opvolgingsrapport uit te dokteren. Gebaseerd op de input van David, had Kenneth het tempo waaraan het projectteam opgeloste fouten hanteert als metric. Maar dat was buiten de waard, of eerder buiten de teamleaders gerekend.

Door wekelijks het aantal “opgeloste fouten” te rapporteren en door er vooral naar te streven dat aantal op te drijven, begon meer en meer duidelijk te worden hoe laag de softwarekwaliteit eigenlijk wel was. Waar vroeger de gevonden fouten werden stilgezwegen, stonden ze nu steeds centraal op de opvolgingsagenda.

Eerst waren de teamleaders bij Kenneth geweest om aan te geven dat kleine fouten, spellingsverbeteringen, veldjes die wat te lang of te kort waren, niet gerapporteerd moesten worden. Dergelijke banale dingen waren het rapporteren niet waard. Kenneth had echter niet ingestemd – een fout is een fout. Als gevolg daarvan was een discussie ontstaan over de definitie van een fout – met de bedoeling om twee categorieën te creëren: fouten en cosmetica. Deze laatste zouden van de rapportering wegvallen.

Eerst had Kenneth ingestemd maar hij stelde al gauw vast dat die opsplitsing extra werk vergde en dat op het einde van de rit het toch wel belangrijk was om te weten welk type fouten het meest voorkwam om zo naar de toekomst toe het ontwikkelingsproces te verbeteren.

Spanning dus tussen Kenneth en de teamleaders.

Eenmaal had hij zèlf een fout gemaakt in het berekenen van zijn resultaten. Hij werd quasi gevierendeeld door de teamleaders. Die zagen hun kans schoon om te proberen van Kenneth en zijn rapporten af te geraken.

Kenneth had uiteindelijk de opdracht te gekregen om zijn rapporten te vereenvoudigen. Gevolg was dan wel dat inderdaad een pak details rond foutrapportering wegviel, waardoor het leek alsof de kwaliteit plots verbeterd was...

Dit zou nog een staartje krijgen.

### 33

Het was het einde van de tweede week, vrijdag. Na het weekend zou de software voor het eerst gepackaged worden en op CD gebrand. Dinsdag was een sanity-check gepland van de CD waarna hij op woensdag eindelijk naar de bètaklanten zou vertrekken.

Kenneth zat te bedenken dat men in de planning er blijkbaar vanuit ging dat de sanity-check positief zou uitvallen en dat er dus geen correctie meer gepland hoefde te worden. Hij wou dit melden maar aarzelde. Hij was al de hele week aan het muggezippen geweest. Tenminste, in de ogen van de ontwikkelaars.

Miriam kwam binnen. Moe. Net een meeting achter de rug met de mannen van het databaseteam. Ook al zijn ze poeslief met haar, het blijft een hele inspanning om elk misverstand te vermijden.

“Ga je mee nog iets drinken”, vroeg ze aan Kenneth. Kenneth keek verbaasd op. Miriam had hem nog nooit uitgenodigd. Hij haar trouwens ook niet.

“En?”, zei ze, “of past het niet”. “Ja, euh, ja, ik heb hier toch ook gedaan, thanks. Ik ben zo bij je”. Kenneth stamelde wat.

Ze gingen naar een café, enkele straten verderop. Een wijkje met allerlei restaurantjes, beetje duur.

“Het gaat goed hé”, begon Miriam, “maandag releasen met geteste software, wie had dat gedacht.”

Kenneth wou haar enthousiasme niet temperen, maar hij was er toch niet zo gelukkig mee. Hij vreesde dat het releasen en packagen maandag problemen zou opleveren.

Hij had namelijk gezien dat het aantal fouten toenam als men een integratie uitvoerde, zij het nu van twee modules of meer, of in dit geval van de software met de installatieprocedure. Die CD-authoring zou wellicht voor onstabiliteit zorgen.

Hij vertelde het toch aan Miriam. “Weet je”, zei hij, “Integratietesten zijn de meest onderschatte testen van een ontwikkeling”. Vaak vindt men dat het overbodige testen zijn, zeker als de modules al eens getest werden en als er nog een algemene systeemtest gepland is. Die systeemtest vangt toch de problemen op mochten die er zijn.”

Miriam keek hem aandachtig aan. Kenneth ging verder. “Een integratietest is een goede barometer voor de toekomstige kwaliteit van het hele systeem. Als bij een integratietest het aantal fouten toeneemt, dan betekent dit dat de software onstabiel wordt naarmate men de verschillende onderdelen assembleert. Dit duidt vaak op een slechte productarchitectuur.

Bekijk het zoals een auto of huis. Je kan heel luxueuze en goede zetels in een kwalitatieve auto monteren, maar als het een om een sportauto gaat, zal je toch door elkaar geschud worden als het geen sportzetels zijn.

Ik heb gezien dat het aantal fouten bij integratietesten toeneemt. En ik weet ook – dat weet jij ook – dat niemand eigenlijk een goed totaalbeeld heeft over de architectuur van

onze applicatie. Ik vrees dus dat het packagen maandag problemen zal opleveren... Er zullen onverwachte problemen optreden als gevolg van de integratie tot één systeem.”

Miriam had afgehaakt. Ze had net haar strawberry margueritta op. “Eten we nog iets?”, vroeg ze.

## 34

Het was laat toen ze uiteindelijk het restaurant verlieten. Ze lachten. Ze hadden een goede avond gehad. Kenneth was verbaasd geweest. Hij had niet verwacht dat Miriam uiteindelijk nog kon meevallen. Op het werk was ze altijd zo genadeloos en ze maakte zo opvallend gebruik van haar aardige verschijning.

Hij was er uiteindelijk ook wel door gecharmeerd geweest deze avond. Ze was gewoon beeldschoon. En natuurlijk, hoe later op de avond, hoe mooier ze er begon uit te zien. Een fles wijn en wat sfeerlicht.

Ze wandelden terug naar kantoor, naar de parkeergarage. Miriam had eigenlijk wel wat teveel op. Ze had haar arm al rond Kenneth geslagen. Onwennig had hij haar dan maar wat ondersteund op weg naar de wagen.

“Zou ik je niet naar huis brengen?” stelde Kenneth voor.

“Dat zou inderdaad verstandig zijn,” antwoordde Miriam, “als je het niet te lastig vind?”

Kenneth zette Miriam voor haar deur af. Ze woonde min of meer in dezelfde richting als waar Kenneth naartoe moest. Aan het station. Ze kon dus op maandag met de trein naar het werk. Of morgen, om haar auto op te pikken.

“Kom je nog mee binnen”, vroeg ze.

“Nee, dank je. Ik ben moe”, schudde Kenneth.

“Ok. Toch bedankt,” zei ze. Ze leunde naar voor, greep Kenneth’s gezicht tussen haar twee zachte en nu warme handen, en zoende hem vol op zijn mond.

Voor hij het besepte was ze uitgestapt en naar binnen gewandeld. Ze wuifde nog even na.

Het waren zachte volle lippen. Hij proefde haar lipstick nog en rook de mengeling van haar zacht parfum en de geur van haar huid.

## 35

Maandag, D-day. Toen Kenneth toekwam, was het packagen al begonnen. Hij kwam Lieve tegen, ze groette hem koel.

“Was er iets?”, vroeg hij zich af.

Op het werkeiland zat iedereen te werken. Alleen Kurt's stoel was leeg. Sedert hij aan de ontwikkelaars was toegekend, zat hij meestal in de open space van de programmeurs. Miriam zat met haar rug naar hem. Kenneth deed moeite om uitdrukkelijk goeie dag te zeggen. Miriam beantwoordde het even koel als elke andere dag. Hij had iets anders verwacht. Een blik van verstandhouding. Een verwijzing naar vrijdag. Niets van dat alles echter.

Lieve kwam terug binnen. “Kenneth, heb je even?” begon ze.

“Er zijn problemen en ik denk dat het alleen erger zal worden”. Ze was wat opgewonden. “De installatiesoftware blijkt van dezelfde DLL<sup>3</sup>'s gebruik te maken als onze software, alleen de versie is anders. Onze software vereist de laatste versie, maar die werkt niet met de installatiesoftware.”

“Heb je al eerder dergelijke problemen gezien? Weet je of er nog andere fouten gerapporteerd geweest zijn hierrond?”

“Ja”, antwoordde Kenneth. “Een tijd geleden is er een bug binnengekomen van ontwikkelaar. Hij had geprobeerd om de software installeren en omdat zijn pc nog met die oude DLL

---

<sup>3</sup> DLL: Dynamic link libraries: Stukjes software die aan het besturingssysteem van een computer worden toegevoegd en die door meerdere andere programma's kunnen gebruikt worden tijdens hun werking.

zat, was het fout gegaan. Hij had het als fout gerapporteerd omdat hij vond dat die oude DLL moest ondersteund worden. Frank – de teamleader – had mij toen echter gezegd dat dit niet waar was, en in die discussie rond het aantal gerapporteerde fouten, heeft hij mij toen gezegd om die fout zelfs weg te halen uit de database. Het was geen fout van de software maar van het operating system.

“Maar ik verwacht dat er nog van die lijken uit de kast gaan vallen”, ging Kenneth door, “Ik denk dat de software onstabiel zal worden met het integreren met de installatieprocedure en de andere authoringsoftware.”

Hij aarzelde even. “Dinsdag releasen, ik vrees er eigenlijk voor”, zei hij.

## 36

“En dat zeg je me nu!”, haalde Lieve uit naar Kenneth. “Waarom heb je dat nooit eerder gerapporteerd? Is de stabiliteit gedaald?”

Kenneth knikte bevestigend. Hij was geschrokken van haar uitval.

Lieve bedaarde. “We moeten iets doen. Iedereen zit te hopen op de die release van morgen. De klanten zitten nu echt te popelen. Er is zelfs een hele campagne aan verbonden.”

Stilte. Kenneth had geen antwoorden.

“Niet packagen”, zei hij plots, “als we gaan packagen, dan is het risico de software ontstabiel wordt, te groot. Installeer de software bij de bèta-klanten zoals we hier een nieuwe versie installeren op de testafdeling; met de hand.”

Hij begon luidop te denken. “Als we nu eens met een ontwikkelaar, een tester en natuurlijk een verkoper, persoonlijk de software bij de klant gaan installeren. We kunnen dit doen vanuit het beeld dat het om een bèta-installatie gaat en dat we onze klanten persoonlijk willen behandelen.”

“En dan kunnen we nu, in plaats van te packagen, ons bezig houden met het voorbereiden van zo'n installatieproces en van het maken van afspraken.”

Lieve vond het een goed idee. “Knappe gast, die Kenneth”.

## 37

De verkopers vonden het veel minder een goed idee. In hun ogen sloeg het nergens op. De klant zou nog meer denken dat we geen software meer kunnen maken. En daarenboven, misschien waren de klanten niet van plan om direct te installeren.

“Reden te meer om het zo aan te pakken”, zei Kenneth, “want dat betekent dan gewoon dat onze deadline om te releasen later komt te liggen. En voor de klant zal het overkomen alsof we hem een plezier doen door op een tijdstip dat hem uitkomt te komen installeren.”

Hij slaagde erin, met de hulp van Lieve, om de verkopers te overhalen. De klanten werden gebeld.

Allemaal waren ze akkoord met de aanpak. Meer nog, slechts eentje had voorzien om deze week nog de bèta-versie in gebruik te nemen. De meesten hadden twee tot zelfs vier weken later gepland. Hoewel ze oorspronkelijk allemaal hadden gezegd dat ze morgen de software moesten hebben.

Twee vliegen in één klap dus. Klantcrisis vermeden en meer tijd om nog te ontwikkelen EN testen.

## 38

Koffie met David.

David was naar het kantoor gekomen. Hij had geluisterd naar het idee om de klanten persoonlijk te benaderen. En hij had bevestigend geknikt toen hij hoorde dat de klanten allemaal hun installatie later in de tijd voorzien hadden.

“Wat je gedaan hebt”, zei David tegen Kenneth en Lieve, “is een stuk van de buffertijd van de klant gebruikt”.

Het was te onduidelijk voor Lieve en Kenneth.

David verklaarde. “In elk project, zeker waar meerdere mensen in zitten, gaan mensen meestal hun eigen werk en eigen deadlines beschermen door bij hun afspraken en tijdsinschattingen wat reservetijd bij te tellen. Voorbeeld: stel dat ik je vraag hoeveel tijd je nodig hebt om de software bij mij te installeren. En stel dat je zeker wilt zijn dat je mij niet teleurstelt. Dan zal je altijd de echt benodigde tijd plus een pak reservetijd zeggen. Als de installatie één uur duurt, zal je makkelijk zeggen dat het in een voormiddag wel geïnstalleerd geraakt.

Dit verschijnsel komt op vele plaatsen voor in een project en blaast uiteindelijk een project qua doorlooptijd op tot meestal het dubbele van de echt benodigde tijd. Maar dan nog lopen projecten uit. Ook al is er die reservetijd voorzien.

Dat komt dan weer, ik leg het nu iets te snel uit, omwille van het student-syndroom. Als je vier dagen krijgt om een taak

te maken, dan zal je die vier dagen opgebruiken. Zelden zal je eerder klaar zijn. Nochtans zal je een deel van die tijd niet aan de taak besteed hebben. Vooral in het begin, ver van de deadline doe je allerlei zaken die er eigenlijk niets of maar weinig mee te maken hebben. Naarmate de deadline nadert, begin je echt aan de taak te werken. Als er dan nog iets mis loopt, wat uiteraard af en toe voorvalt, dan heb je weinig of geen reservetijd meer om dat op te vangen want je hebt je reservetijd, je buffer, al opgebruikt met allerlei minder relevante taken. Je project loopt dus direct uit. Projecten krijgen eigenlijk veel te veel tijd, niet te weinig.

Lees eens Critical Chain van Goldratt, daar staat dat allemaal mooi uitgelegd,” besloot David., “theory of constraints toegepast op project management.”

“Nu, hoedanook een goed idee om je klanten te persoonlijk te contacteren en met hen een planning te maken. Nu alleen nog zorgen dat jullie voldoende fouten vinden en herstellen.”

## 39

Allemaal goed en wel. De klanten hebben wel uitgestel toegelaten maar wat echter niet zal opgelost geraken, is die afnemende stabiliteit als gevolg van dat integratieprobleem.

Het Hoofd was opgetogen geweest met het uitstel. Hij vond dat het project gered was. Door hem natuurlijk. Dat het idee van Kenneth kwam, was irrelevant. Hij had ervoor gezorgd dat de beste programmeurs werden vrijgemaakt op bij de klanten te gaan installeren. Er hoefde dus niets meer te veranderen.

Hij had geen oren naar het integratieprobleem. Er was nu toch meer tijd, dus konden de testers er maar voor zorgen dat alles opgevangen werd.

Lieve en Kenneth waren het er niet mee eens. Ze zaten samen te overleggen.

## 40

Het espressomachien was stuk. Geen lekkere koffie. Kenneth stond te prutsen aan de watertoevoer. Lieve was half luidop aan het denken, half aan het praten tegen Kenneth. Achter haar stond een programmeur te wachten. Hij wou ook een espresso. Hij ving flarden van het gesprek op.

“De programmeurs moeten testen”, zei hij. Lieve draaide zich om en zag hem. Het was Jacques. Eén van de seniorprogrammeurs, ttz., hij zat al vier jaar bij Beetleware. Stille jongen. Speelt op met de duiven. Maakt hele goeie software. “Minder programmeren en meer testen”, herhaalde hij.

Lieve keek sceptisch. “Neen,” zei ze, “we hebben net vastgesteld dat we al meer fouten gevonden hebben, dan er kunnen hersteld worden. We moeten, en we zijn ondertussen, gestopt met testen.”

“Ik bedoel,” ging Jacques verder, “dat de programmeurs moeten testen, niet de testers. De programmeurs moeten meer testen voor ze hun stuk doorgeven. Pas als hun deel voldoende stabiel is, mag je aan integratie denken.”

Lieve en Kenneth waren bijzonder aandachtig.

Jacques vervolgde: “Jullie zijn altijd bezig met jullie testmethodes en zo. Ik heb het daar allemaal niet zo op begrepen. Want jullie testen als het al te laat is. Jullie testen net voor we willen releasen. Ervoor kan niet, want jullie kunnen de software dan nog niet laten werken.

Als ik software schrijf, dan spendeer ik ongeveer 10 tot 20% aan het coderen, het echte programmeren. Al de rest van de tijd, 80%, gaat naar het nakijken van wat ik gemaakt heb. Ik installeer de code, controleer het op allerlei manieren en haal er nog een pak fouten uit. Zelden echte programmeerfouten, meestal zaken die ik vooraf niet kon weten.

Een goeie programmeur schrijft meestal goeie code. Maar meestal weet de programmeur maar half waar zijn code terecht komt, in welke applicatie, misschien een andere module, welk operating system enzovoort. Dat is quasi nooit gedocumenteerd.

Heel vaak is de goeie, juist gemaakte code, niet compatibel met de omgeving waar het zal in terecht komen. Dat vind ik met mijn testen. Dan moet ik mijn code aanpassen, zodat het wel past in het geheel, met behoud van de juiste functionele werking natuurlijk. Meestal moet ik bepaalde routines anders aanpakken, andere klassen gebruiken of zo. Er zijn veel manieren om één stuk code te schrijven. Zelden is de eerste manier de juiste, of beter gezegd, de bruikbare.”

## 41

Interessant verhaal. De programmeurs moeten dus meer testen. Vroeger deden ze dat nochtans redelijk veel. Dat was trouwens een opmerking van die audit die ooit gedaan was. Er was geen professioneel ontwikkelproces. Iedereen zat op zijn eilandje te werken, programmeren en testen door elkaar.

Nu was er die nieuwe aanpak, met een heus testdepartement. En inderdaad, er was al vastgesteld dat de programmeurs nu de neiging hadden om veel minder te testen. “Ze gooiden hun software over het muurtje”, werd er vaak gezegd. Er waren nu toch testers, dus waarom dubbel werk doen, en nog zelf testen.

Zou die nieuwe aanpak de oorzaak van alle problemen zijn?

Lieve zat te denken. Hoe kreeg ze dat aan het Hoofd verkocht?

## 42

Ze was naar de Baas gestapt. Het Hoofd zou toch niet luisteren, laat staan akkoord zijn, had ze haar de bedenking gemaakt.

De Baas luisterde. “Zit iets in”, zei hij. “Heb je het er al over gehad met Development?”, vroeg hij.

“Nee”, ik vrees dat ik niet gehoord zal worden, “het is immers geen prettig nieuws. En als het daarenboven van mij zal komen, zullen ze zeggen dat ik zelf mijn werk maar moet doen. Ik ben uiteindelijk voor het testen verantwoordelijk.”

“Ik heb de statistieken van de fouten per module”, zei de Baas, “en daar kan je inderdaad op zien dat de software van Jaccques weinig problemen vertoont. Dit is het beste bewijs dat de methode van Jacques werkt. Ik zal eens met development praten en hen zeggen dat ze Jacques moeten volgen. Jacques zullen ze daarenboven meer aanvaarden dan jij natuurlijk. Hij is één van hen.”

Lieve knikte instemmend. Ergens voelde ze zich een beetje een lafaard. Ze was achter de rug van haar collega om, naar hun gemeenschappelijke baas gestapt. Niet echt de beste methode om een goeie collegiale relatie uit te bouwen.

Aan de andere kant, ze zag het sowieso niet zitten om nog met het Hoofd een goeie relatie uit te bouwen.

Relaties trouwens. Ze liep nog steeds met Kenneth in haar hoofd. Ze was bijzonder onder de indruk geweest van zijn idee om manuele bèta-releases te organiseren. Ze had

zich slecht gevoeld omdat ze zo was uitgevallen tegen hem. Alsof het zijn schuld was dat er zoveel fouten in de software zaten.

Ze is zelf altijd kwaad als men de testers met de vinger wijst als er fouten bij klanten gevonden worden. Wie had ze er immers in gestopt...

## 43

Het bedrijf is veel te veel in twee kampen verdeeld. Programmeurs versus testers. Dat is niet goed. Dat is ook een beetje het gevolg van die nieuwe aanpak. Die externe consultants hadden immers aangedrongen op het opzetten van een apart testdepartement. Aanvankelijk zelfs met externe testers. Van hun bedrijf natuurlijk.

Dat had gelukkig geen weerklank gevonden. Dat consultantsbureau mocht wel adviseren bij het opzetten van het testproces en het recruter en opleiden van de testers, maar het moesten wel allemaal eigen mensen worden. Geen externen. Die waren trouwens veel te duur.

Maar toch waren er twee kampen. En de verzuurde relatie tussen het Hoofd en haarzelf versterkte dat beeld alleen maar.

Misschien toch meer naar een integratie streven? Maar dan kwam haar eigen job in het gevaar? Immers, als de testers worden toebedeeld aan het ontwikkelteam, waarom zou je dan nog een teamleader testing nodig hebben?

## 44

De eerste handmatige bèta-installatie was achter de rug. Het was allemaal niet vlot gelopen, maar bon, in twee dagen tijd was de installatie gelukt. Het systeem van de klant was gedurende die twee dagen wel bijna voortdurend out geweest, maar had men gewoon een cd opgeleverd, dan zou het veel langer geduurd hebben, en belangrijker, de klant zou veel bozer geworden zijn. Twee dagen was hoedanook veel meer dan de voorziene twee uur die ze hadden aangekondigd.

En als klap op de vuurpijl had de installatie van het servergedeelte van de software ervoor gezorgd, via de auto-update procedure van de werkstations, dat dertig van achtendertig pc's waren gecrashed over een periode van vijftien minuten. Gelukkig dus dat iedereen onsite was en dat de installatie tijdens een weekend doorging. Tegen de maandagochtend was alles terug in orde. Veel hadden ze niet geslapen de laatste achtenveertig uur.

Kenneth was mee geweest naar die eerste installatie, als de tester van het team.

Hij zat wat na te praten met Lieve. Het was weer avond. Het regende pijpenstelen. Verkeersellende tot gevolg. Dus nog maar wat op kantoor blijven.

“Best dat we daar met vier man waren”, zei Kenneth, “zeker met onze database- en Operating System-expert. We hebben toch een pak fouten gevonden, die we hier nooit zouden kunnen hebben gevonden. Ik was eigenlijk geschrokken van hun omgeving. Onze testomgevingen

benaderen in geen jaren wat die klant daar als configuratie heeft.”

“Veel te duur”, zei Lieve, “je kan onmogelijk elke klantsituatie hier simuleren. En als je het zou kunnen, dan nog zou het niet realistisch zijn. De omvang van de installatie heb je hier niet, ook de netwerkbelasting niet.”

“Ja, maar dat is niet echt wat ik bedoel”, repliceerde Kenneth, “we wisten gewoon niet hoe de situatie van de klant ineen zat. Welke pakketten hij nog allemaal gebruikt. En vooral, hoe ze de software gebruiken. Ik heb daar users gezien die bijvoorbeeld de terminal editor die in Windows zit als een clipboard gebruiken, om zo klantenummers van de ene naar de andere applicatie door te geven. Als wij dan, die terminal editor updaten en aanpassen voor onze applicatie, dan is het normaal dat de gebruiker het noorden verliest.”

“En,”, ging hij verder, “ik wil ook niet zeggen dat die gebruiker correct tewerk gaat op die manier, daar kan je trouwens uren over filosoferen, ik bedoel vooral dat we het niet weten. We weten niet wat we niet weten. Dat zou eigenlijk niet mogen kunnen.”

Lieve knikte instemmend. We weten niet wat we niet weten. Was dat wel het geval geweest, dan hadden we op zijn minst kunnen realistischer kunnen inschatten wat de mogelijke problemen waren. Hoe lang was het geleden dat het ontwikkelingsteam, testers inbegrepen, nog een klant hadden bezocht. Er zaten zeker programmeurs bij die nog nooit met een klant hadden gesproken. Ze kenden de requirements alleen uit een dossier.

Kenneth kende de klanten redelijk goed. Toen hij op support zat, had hij vele malen interventies mogen uitvoeren. En aan de telefoon had hij ze ook heel goed leren kennen. En één ding was zeker, ze gebruikten de software allemaal op een manier die je je amper kon inbeelden, zeker als de bouwer van het product bent.

## 45

Ze besloten naar huis te gaan. Lieve praatte wat over een dvd die ze net met een spaarbon uit te krant had gekocht. Hij kende de film wel. Had ze gemist toen hij in de bioscoop liep. Of hij geen zin had om te komen kijken?

Waarom niet. Kenneth volgde Lieve door het drukke stadsverkeer. Het regende nog steeds.

Lieve woonde een goeie veertig kilometer verder op. Op het platte land. Haar dochter was er niet. Ze was die week bij haar vader. De hond was er wel.

Een beetje onwennig ging Kenneth de huiskamer binnen. Hij nam de kamer in hem op. Warme kleuren, wellicht enkele jaren geleden al ingericht, niet te trendy. Oranje en bordaux hadden de boventoon in het salon. Hij zag wat blauw en wit uit de keuken piepen. Lieve ging naar achter om de hond te begroeten die op het terras zijn plek had. Geen honden in huis. Vond hij wel goed.

Een grote lange grenen eettafel, achter het salon.

Een huis van vrouwen, dacht hij. Je zag aan de aankleding, de accessoires en natuurlijk de magazines in de krantenbak dat er geen man woonde. Veel kaarsen maar geen thee-lichtjes. Ook goed, vond hij. Hij had een hekel aan de steriele alomtegenwoordige witte overmaatse pastilles die in elk ornament voor gezellig kaarslicht moesten zorgen. Een opdringerig symbool van een gestandaardiseerde knusheid. Veel beter waren de gewone kaarsen, in verschillende lengtes, diktes en kleuren. Hoewel hij zelf geen kaarsen brandde.

Met wat vragen over de hond bemeubelde hij de stilte, die hij wat onbehagelijk begon te vinden. Hij hield zich veilig op voor de hand liggend terrein. Geen waardeuitspraken over het interieur. Hoe oud de hond was? 7 jaar. Hoe lang ze hem had? 6 jaar, uit een asiel. Hoe hij heette? Het was een zij. Frits. Rare naam voor een vrouwtje – teef vond hij zo'n verschrikkelijk woord.

Lieve kwam terug binnen. Frits had haar portie aandacht en eten gekregen. Ze stond kwispelend Kenneth aan te kijken. Achter het vliegenraam.

“Eten?”, vroeg Lieve.

## 46

Gemengde sla met spekjes en drie soorten kaas, feta, gouda en mozzarella, bestrooid met wat cocospoeder. Multicultureel. En jagersbrood. Zo'n bruin groot plat vierkant brood. In de supermarkt hadden ze heel lekker jagerbrood. Een fles witte wijn. Droge.

Zelf met de regen op de achtergrond was het zomers eten. Het deed Kenneth denken aan zijn tijd in Zuid-Frankrijk, waar hij geregeld, maar dan buiten, dergelijke slaatjes at met zijn klanten.

Lieve verdween even. "Lenzen uitdoen", zei ze, "op het einde van de dag irriteren ze soms. Ik ben zo terug. Maak het je gemakkelijk. De DVD ligt op de tafel."

Kenneth ruimde wat af. Stapelde alles in de keuken op het aanrecht, daar waar hij dacht dat het wel mocht staan. Het was er netjes.

Lieve kwam terug binnen, Kenneth keek op. De lenzen hadden plaats gemaakt voor een bril. Een kleine, hoekig van vorm. Het flatteerde haar gezicht bijzonder. Zo had hij Lieve nog nooit gezien. De potloden waren uit het haar verdwenen. Haar tot op de schouders. Ze zag er tegelijkertijd jonger en toch wat ernstiger uit. Maar heel verleidelijk.

"Leuke bril", zei hij. Dan, na wat aarzeling, "het staat je, heel mooi". Lieve glimlachte. Een compliment is natuurlijk altijd leuk, maar ze hoorde duidelijk dat een oprechte spontane, of toch bijna spontane, reactie was.

“Wil je wat koffie, of drinken we eerst de fles wijn leeg?” vroeg Lieve. Raar, hoe zelfs volwassenen zich aanstellen en niet gewoon zeggen wat ze bedoelen.

De wijn verhuisde naar de salontafel. De dvd werd opgezet. Honderdtwintig minuten film nog te gaan.

**47**

Het werd laat. Ze waren uiteindelijk na achten van het werk naar huis gereden. Kwart voor middernacht. De film was gedaan. Kenneth stond op. “Ik ga er maar eens vandoor”, zei hij. De toon klonk alsof hij een bevestiging of zo wou horen. Op National Geographic zou men nu zeggen dat de paringsdans begon.

Lieve begeleidde hem naar de deur. Een soort spannende sfeer, die al geruime tijd in de kamer had gehangen, werd nog meer voelbaar. Kenneth was nooit een iemand van kordaat handelen geweest, zeker niet als het op vrouwen aankwam. Durfde zelden initiatief nemen.

Onhandig naderden ze elkaar en kusten ze, kort. Het leek een afscheidskus, maar het was veel meer een verborgen test van beide partijen om te kijken in hoeverre ze méér wilden.

Ze wilden meer. Het duurde onnatuurlijk lang voor ze wat afstand namen. Genoeg als signaal om het nog eens over te doen. Maar dan goed.

De voordeur ging niet meer open.

## 48

De ochtend was natuurlijk vreemd. Kenneth moest eerst nog naar huis. Verse kleren aantrekken, zich scheren. Hij stond vroeg op, kleepte zich aan. Lieve lag nog knus onder de lakens, half te soezen. Een witte donsovertrek. Heel sereen, geen trendy design. Had hij mooi gevonden. Hij boog over haar heen, kustte haar. Ze omhelsde hem. Het voelde goed. Ze voelden dat allebei.

Breed glimlachend stapte hij in zijn auto. Hij keek naar zichzelf in zijn achteruitkijkspiegel – toch een beetje macho.

Onderweg moest hij onwillekeurig denken aan zijn avond met Miriam. Miriam was toen vrijpostig geweest, door de alcohol. Dit was anders.

Thuis, terwijl hij zich schoor, genoot hij nog van gedachte van Lieve, die dicht tegen hem aanlag, warm en zacht. Hij had nog naar haar liggen kijken toen ze in slaap gevallen was. In de schemering van het licht dat er niet was, zag hij de zweetdruppeltjes op haar voorhoofd en hals glinsteren. Amper zichtbaar in het donker, maar hoor- en voelbaar, haar adem tegen zijn borst. Het cliché wil dat mannen altijd direct in slaap vallen, maar daar was Kenneth te opgewonden voor geweest.

Het was lang geleden dat hij nog bij een vrouw had geslapen.

## 49

Er waren vier bèta-releases achter de rug. De problemen van de eerste installatie had zich in beperkte mate nog eens voorgedaan bij de tweede installatie. Maar daarna niet meer. Ondertussen was er ook wel het een en het ander verbeterd. De laatste bèta was al een totaal andere versie dan de eerste, v5.4.1b3. Het versienummer werd alleen maar langer. De lengte van het versienummer correleert meestal omgekeerd met de kwaliteit van de software.

Een aantal van de programmeurs, de meeste eigenlijk, waren ondertussen zover dat de methode van Jacques wilden volgen: een beetje programmeren en dan veel testen. Unit-testen, zoals dat heet. Sommigen experimenteerden met het uitwisselen van software om te testen onder de programmeurs. Maar dat werkte niet altijd. Het vergde een bijkomende coördinatie. De één zat dan op de andere te wachten.

Hoedanook, ze vonden zelf veel fouten. Het feit dat ze hun fouten zelf vonden, maakte daarenboven, dat ze dat hoge aantal aan fouten, minder schokkend vonden, tzt. minder beledigend voor henzelf.

De laatste bèta was stukken stabielier dan de eerste. Er deden zich nog wel wat integratieproblemen voor. Die waren vooral het gevolg van enkele architecturale kwesties in Sigma. Die waren in dit stadium niet meer te herstellen.

Ze waren eindelijk zover dat ze wisten over welke delen van de software ze eigenlijk niets wisten. Zo kon men veel beter de installatieproblemen inschatten.

Ondanks het feit dat de vierde bèta succesvol leek te zijn, was er toch nog een pak schroom en miserie om nu met een afgewerkte release voor de dag te komen.

Een aantal klanten, waaronder de bètaklanten, had ondertussen op basis van hun testervaring een pak nieuwe features gevraagd en nu was de vraag of deze niet meteen ook moesten ontwikkeld worden. Men was er nog niet uit.

## 50

Het werd zomer, al was het pas eind mei.

Sigma had zijn officiële release nog steeds niet gekend. Bèta5 was uit en men had alle bètaklanten geupdate. Dat was echter nooit de bedoeling geweest.

Het management begon te morren en de Baas lag onder vuur. De CEO had hem al een paar keer naar zijn kantoor gesommeerd. Vanuit marketingstandpunt begon het erop te lijken dat Beetleware geen software meer kon leveren.

Nieuwe klanten waren er ondertussen de laatste drie maanden niet bijgekomen. De prospecten die in januari software hadden verwacht, hadden ondertussen afgehaakt.

Intern begon het ook te rommelen. Het ontwikkelingsteam begon Sigma als een last te zien, een doembeeld dat hun leven beheerste. Er hing een loodzware sfeer.

De relatie met de testers was er nog niet veel op verbeterd. De testers zaten trouwens ondertussen terug samen. Kurt was voor de tweede maal gedemotiveerd. Hij had gehoopt dat zijn tijdelijke verhuis naar development een permanent karakter zou krijgen. Niet dus.

Het Hoofd was weer arrogant en onbeleefd zoals voorheen. Sommige dingen veranderen gewoon niet.

De Baas had geopperd om een ervaren consultant in huis te halen om de prioriteiten te helpen bepalen en om aan te geven hoe men het snelst de kwaliteit kon verbeteren.

De CEO had een eenvoudige vraag gesteld: “Hoeveel externe consultants heb je nog nodig om je te vertellen wat zou moeten weten? En waaom zal die externe consultant het wel weten en jij niet?” Twee vragen eigenlijk.

De Baas gaf toe, vooral aan zichzelf, dat hij eigenlijk al goed kon voorspellen wat die consultant zou vertellen. Het mocht dan nog een oude schoolvriend zijn. De consultant zou bevestigen wat de Baas al wist.

Consultants dienen vaak als schaamlapje. Ze worden ingeschakeld om beslissingen uit te stellen onder het mom van gedegen onderzoek. Of om beslissingen die men al genomen heeft in de geest, bevestigd te krijgen. Altijd een self-fulfilling profecy natuurlijk.

Geen consultants dus. Wel een lunch met de schoolvriend, die inderdaad niets fundamenteel baanbrekends in te brengen had. Alleen, de schoolvriend kon het verhaal dat hij aanhoorde, helder synthetiseren. Dat is sowieso moeilijk voor wie midden in het probleem zit. Wie vanuit het dal naar bergbeklimmers kijkt, ziet immers veel beter hoe een team aan mekaar hangt.

En de consultant bevestigde, het Hoofd moest eruit.

## 51

De Baas had een elegante oplossing gevonden. Hij was enige tijd gelede nog naar een voordracht geweest over CMM, het Capability Maturity Model, een model dat vertelt hoe professioneel je bent in het maken van software, zeg maar.

Daar had hij geleerd dat bedrijven in Europa én de VS inspanningen deden op het niveau twee of drie te halen op een schaal van vijf. In India daarentegen, in Bangalore, stonden veel off-shore softwareontwikkelingsbureau's op het hoogste schavotje, niveau vijf.

Hij wou al enige tijd met het CMM-model de softwareontwikkeling veranderen. Hoewel hij zijn twijfels had of het veel zou uithalen.

Maar nu wist hij het zeker. Het zou niets uithalen zolang het Hoofd de plak bleef zwaaien in het ontwikkelingsdepartement. De Baas besloot dus een CMM-project op te starten, veel minder omdat hij er een fundamenteel geloof aan hechte, dan wel omdat hij daardoor het Hoofd een andere plek in de organisatie kon geven.

Het Hoofd werd projectverantwoordelijke CMM, in plaats van algemeen developmentverantwoordelijke. Het projectteam CMM bestond momenteel slechts uit één persoon, maar dat maakte niets uit. Het CMM-project kreeg het adjectief "strategisch" mee, wat de pil moest vergulden die het Hoofd te verteren kreeg. Hij verloor zijn lijnverantwoordelijkheden.

Tegelijkertijd werd er gereorganiseerd. De testers en de ontwikkelaars werden dan toch samengezet in één geïntegreerd departement dat rond het nieuwe ontwikkelingsproces werd georganiseerd. Kenneth had dat proces helemaal uitgetekend, samen met Lieve.

Het werd zijn laatste wapenfeit want Kenneth had uiteindelijk besloten om het bedrijf te verlaten. Zijn initiële overplaatsing naar Sigma had hij nooit een goede oplossing gevonden. En nu hij een relatie had met Lieve, vond hij dat hij niet langer kon werken in dezelfde afdeling.

Het management had trouwens al dergelijk commentaar laten vallen. Aanvankelijk hadden Lieve en Kenneth het stil gehouden, maar uiteraard was dat niet uit te houden op lange termijn. Toch niet zonder enige vorm van bedrog te laten groeien.

Lieve en Kenneth waren dus een stel, en het bedrijf wist dit.

Kenneth had besloten zich zelfstandig te vestigen. Als IT-professional, teamleader of zo. Hij zou zijn aanpak en visie op softwareontwikkeling uitschrijven en proberen aan de man te brengen.

## 52

Lieve werd het nieuwe hoofd, maar dan nu van het hele team. Van de analisten, programmeurs, testers. Er waren nog twee teamleaders uit de organisatie verwijderd. Figuren die teveel en te lang het voorbeeld van het Hoofd hadden gevolgd. En daar had men geen special project voor gevonden.

Het Hoofd was cynisch, zoals altijd, maar kon geen zware schade aanrichten. Hij zat veilig apart in zijn eigen cel.

## 53

Sigma was eigenlijk een verloren zaak. Men verkreeg geen betere softwarekwaliteit dan die van bèta5. De integratieproblematiek bleef de bottleneck, meer specifiek de productarchitectuur.

Kenneth had aan de Baas en Lieve geopperd om de bestaande klanten die dringend een nieuwe versie nodig hadden, de bèta5 te leveren en die met een speciaal bètateam te ondersteunen voor enkele maanden. Het gros van de andere klanten zou op de oude versie blijven werken. Er zouden wellicht wel enkele klanten afhaken. En...hij stelde voor, eigenlijk ingefluisterd door Jacques, om een klein ontwikkelingsteam samen te stellen dat quasi vanaf nul een nieuwe Sigma zou bouwen. De hele zomer lang. Jacques was net in mei op vakantie geweest. Hij kon dus doorwerken.

Het leek een wanhoopsdaad. Een laatste kans voor Beetleware om met de contactmanager nog iets te bereiken. Jacques mocht zijn team samenstellen. Aanvankelijk wou hij Kenneth erbij maar deze had net zijn ontslag aangekondigd.

Een team van acht man werd samengesteld. Vooral enkele jonge programmeurs, briljante gasten, die echter tot voor kort niet echt gedisciplineerd gewerkt hadden. Wat ervaren analisten. Kurt zat er ook bij.

Daan ook. En dat was vreemd. Daan straalde nooit enige dynamiek uit, testte nine-to-five, en dan nog niet eens grondig. En nu plots, nu hij in het Sigma2-team, zoals ze werden genoemd, zat, was hij een vat vol energie. Hij was

bijzonder geëngageerd, enthousiast. De reorganisatie had voor hem blijkbaar op de juiste knoppen geduwd.

Een bètaklant had zich aangeboden om mee te helpen bij de requirementsdefinitie. Belangeloos. Een klant die Beetleware genegen was.

Miriam kwam terecht in het supportteam van de bètaklanten. Ze stond voornamelijk in voor de regressietesten en voor het registreren van nieuwe requirements. Met haar economische achtergrond, kon ze vlot met de klanten praten.

Ze hadden twee maanden om een nieuw pakket te schrijven. De kans op slagen was miniem.

## 54

Ze zaten allemaal in “Den Bloeienden Wyngaerd”. Kenneth gaf zijn afscheidsdruk. Hij was uiteindelijk maar zes maanden bij Sigma geweest, maar toch goed geïntegreerd. Zijn rol als verborgen architect van vele beslissingen had er toch voor gezorgd dat iedereen hem kende en uiteindelijk toch wel apprecieerde. De Baas was er ook. Het Hoofd was er even geweest, maar was na een acte de présence terug vertrokken.

David was er. Met een Blanche in de hand was hij in een hevige discussie getreden met de Baas. Over de praktische toepasbaarheid van de bottlenecktheorie.

En natuurlijk was er Lieve. Niet alleen was het een afscheidsdruk, het was ook een beetje de kick-off van het Sigma2-team. Moed in drinken voor een taak die bijna onmogelijk leek. Het bracht iedereen in een roes. Er werd veel gelachen. Een ontlading na een maandenlange depressieve sfeer.

Het was schitterend weer. Een zonovergoten binnenterras van de taverne. Warm. Van tussen de menigte keken Lieve en Kenneth mekaar aan. Ze lachten. Lieve’s ogen straalden door haar bril, die ze nu af en toe droeg. Kenneth zag opnieuw de glinstering van zweetdruppeltjes, dit keer in de felle zon. Als parels sierden ze haar hals.

Ze was mooi.

Einde.