Case study 'Implementatie Ariane 5'
De ethische aspecten

Inleiding

In dit essay worden de ethische aspecten van het implementatietraject van een *safety-critical system* geëvalueerd. Een *safety-critical system* wordt gedefinieerd als [1]: *a safety-critical system is a computer, electronic or electromechanical system whose failure may cause injury or death to Human beings"*.

In de afgelopen decennia heeft de ontwikkeling van de soft- en hardware een stormachtige ontwikkeling doorgemaakt. In de jaren '80 is het zelfs mogelijk geworden om ICT in te zetten foutgevoelige situaties. Dat lijkt in eerste instantie geweldig, aangezien computers nooit fouten maken waar mensen deze wel kunnen maken. Dus, zo was de redenering, moeten we toe naar *safety-critical systems* om zo mensen te ondersteunen in lastigere, 'gevaarlijke' situaties.
Toch kwam al snel het inzicht tot stand dat dit niet zo gemakkelijk en eenvoudig is als het lijkt. Immers, fouten kunnen ook voorkomen in de soft- en hardware. En wanneer deze ICT ingezet wordt in dergelijke situaties, kan dit leiden tot verwondingen of zelfs de dood van mensen. Daarom is het zeer belangrijk om de verschillende ethische aspecten en verantwoordelijkheden vast te leggen.

In de ruimtevaart spelen deze *safety-critical* systemen uiteraard ook een belangrijke rol. Zo is het besturen van een ruimteveer is een veel te ingewikkelde materie om deze door mensen uit te laten voeren. Deze en andere zaken zijn daarom overgenomen door ICT systemen, om zo het ruimtevaart veel nauwkeuriger en efficiënter te laten functioneren. Een fout in deze systemen heeft echter wel tot gevolg dat deze ruimtemissies catastrofaal aflopen.
Een bekend voorbeeld hiervan is de Ariane 5 raket. Bij het lanceren van deze raket ontstond er een softwareprobleem waardoor de gehele raket ontplofte. In deze case zijn er, gelukkig, geen menselijke slachtoffers gevallen, maar deze case geeft wel aan welke gevolgen ontwerp- en ontwikkelbeslissingen hebben.

De (ethische) vraag die in dit essay beantwoordt zal worden is de volgende: "Moeten software engineers zich houden aan (ethische) richtlijnen om hiermee schade te voorkomen in *safety-critical systems*? "

In het vervolg van dit essay zal deze case worden toegelicht en zal hierover een ethisch oordeel gegeven worden. Dit oordeel zal met behulp van de *impact grid* [2] opgesteld worden.

Case beschrijving 'Ariane 5'

Inleiding

Het *European Space Agency* (ESA) is verantwoordelijk voor de ontwikkeling van de ruimtevaart in Europa. In deze verantwoordelijkheid moet de ESA ervoor zorgen dat via ruimtevaart nieuwe diensten en producten beschikbaar komen voor de inwoners van Europa. Daarnaast ontwikkelt ESA technologieën en diensten welke gebruik maken van satellieten, en bevordert ze de concurrentiepositie van de Europese industrie. ESA werkt nauw samen met ruimtevaartorganisaties buiten Europa, zodat veel van de wetenschappelijke vooruitgang ten bate kan komen aan de hele wereld.
In de context van deze activiteiten voert het ESA ook het Ariane programma uit. Door de ontwikkeling van dit (onbemande) ruimtevaartuig kunnen er tal van experimenten uitgevoerd worden in de ruimte. Het Ariane programma heeft de afgelopen decennia een aantal versies van de Ariane raket opgeleverd. Na de, succesvolle, Ariane 4 raket moest de Ariane 5 raket, een verbeterde evolutie, het mogelijk maken om satellieten in de ruimte 'af te leveren'. Tijdens de eerste testvlucht ging het echter helemaal mis. Op een hoogte van 3700 meter, verliet de raket de oorspronkelijke koers, brak in meerdere stukken uit elkaar en explodeerde. Gemaakte kosten? 500 miljoen dollar voor de vernietigde raket en 7 miljard dollar aan ontwikkelkosten. Uiteraard werd er een groot onderzoek ingesteld naar de oorzaak van dit falen, aangezien de politieke en sociale bereidheid en geloofwaardigheid op het spel stond van het ESA project.

Onderzoek

In het onderzoek hadden de onderzoekers de beschikking over de telemetriegegevens tot 42 seconden na 'lift-off', radarinformatie en teruggevonden brokstukken. Uit dit onderzoek kwamen naar voren dat de eerste 36 seconden na de start normaal waren en dat daarna de hoogte- en stuurinformatie ontbrak. Dit probleem werd veroorzaakt door het *Inerial Reference System* (SKI) welke deze informatie moest verzorgen. Aangezien dit component zeer belangrijk is, dit component levert namelijk de primaire informatie voor het aansturingproces, is dit component door ESA zeer uitvoerig getest, gemodelleerd en redundant uitgevoerd in de raket. Toch was het mogelijk dat er een fout optrad. Bij het berekenen van de *horizontal bias* (HB) ontstond er een conversiefout ontrent de HB, de 64 bit floating point variabele moest in een 16 bit integer getypecast worden, welke niet werd afgevangen door de software. Deze error zorgde ervoor dat de software vastliep waardoor het tweede SKI component dezelfde berekening moest uitvoeren. Uiteraard ging dezelfde handeling ook in het tweede component verkeerd waardoor alle systemen in de Ariane 5 vastliepen en de *self-destruction* werd gestart.

De desbetreffende software was overgenomen uit de succesvolle Ariane 4 raket. Tijdens die ontwikkeling is de software uitvoering getest en goed bevonden. Ook in de vele vluchten is er nooit een fout opgetreden. De systeemontwikkelaars zijn er vanuit gegaan dat deze software dan ook één op één overgezet kon worden naar de nieuwe raket. Hierbij is nooit gedacht aan de mogelijkheid dat de nieuwe raket compleet andere waarden voor de HB kon aannemen dan in de Ariane 4. Er is dus sprake van een specificatie-, ontwerp- en testfout.

De hele testprocedure voor de software van de Ariane is vooraf vastgelegd en bestaat de volgende aspecten:
- Equipment qualification;
- Software qualification;
- Stage integration;
- System validation tests.

Er kan zeer uitvoering worden ingegaan op deze testprocedures, maar feit is dat de gehele configuratie in de software nooit volledig getest is. Dit dus mede door het feit dat formeel was aangetoond dat deze exceptie nooit zou kunnen voorkomen, in de Ariane 4!

Tot slot een belangrijke conclusie welke uit dit onderzoek naar voren is gekomen. Er is geen sprake van een programmeerfout, maar van een specificatie / requirements[1] fout. Immers, de software functioneerde precies naar gelang de specificaties. Dat de specificaties veranderd hadden moeten worden is echter niet in het requirements proces naar voren gekomen.

Aanbevelingen onderzoek

Uit het onderzoek zijn ook een aantal aanbevelingen gekomen. De belangrijkste aanbevelingen worden hieronder kort toegelicht:
- Ontwikkel een nieuwe testfaciliteit waarin er met zoveel mogelijk echte apparatuur getest kan worden. Complete simulaties en coverage[2] tests moeten uitgevoerd worden.
- Voer voor elk software-item een requirements review uit. De uitkomst moet centraal bekend zijn. Zodoende is bekend wat een component kan en wat niet.
- Voer statische analyses uit. Zodoende moet bepaald worden welke aannames er gemaakt zijn en welke variabelen er gebruikt worden (vooral de range van deze variabelen).
- Ontwikkel een geheel nieuw testprogramma, met daarin nieuwe test- en verificatieprocedures, om zo tot kwalitatief betere software te komen.

Geconcludeerd kan dan ook worden dat het testprogramma te wensen over liet. ESA heeft deze voorstelling overgenomen en nieuwe ontwikkelprocedures ontwikkeld. Daarnaast is, misschien verrassend, niemand aangeklaagd wegens deze fout.

In het volgende gedeelte van dit essay zal er ingegaan worden op de ethische aspecten en verantwoordelijkheden van alle stakeholders.

[1] Uit [1 blijkt ook dat bij safety-critical systems meestal requirements fouten ten grondslag liggen van fouten
[2] Testen hoeveel coderegels er uitgevoerd zijn tijdens een test. Dit om 'dode code' op te sporen.

Ethische analyse

In dit gedeelte van het essay worden de ethische aspecten geanalyseerd. Dit zal gedaan worden door een stakeholder analyse [3], een analyse over de doodzonden welke voorkomen moeten worden bij *savety-critical systems* [4], een analyse over de mate waarin de *code of Ethics* voor ontwikkelaars zijn toegepast [4], en de *impact grid* [2].

Stakeholders

In de case zijn er een groot aantal stakeholders te onderkennen. De belangrijkste stakeholders zijn de *ESA*, de lidstaten van de ESA, ontwerpers, behoefteanalisten en testers. De programmeurs worden buiten beschouwing gelaten aangezien zij hun werk goed gedaan hadden: de software was geschreven voor de Ariane 4 en was kwalitatief goed. Hieronder worden de stakeholders, de bijbehorende acties en mijn mening hierover kort toegelicht.

Engineers (ontwerpers, behoefteanalisten en testers)
De ontwerpers, analisten en testers zijn verantwoordelijk voor het ontwerpen en testen van de componenten om zo tot een goed product te komen. In deze case hebben deze engineers het nagelaten om legacy componenten niet opnieuw te testen in hun nieuwe omgeving. Het is algemeen bekend in de ICT wereld dat gehele systemen en niet alleen de afzonderlijke componenten getest moeten worden. De verschillende stappen in het ontwikkelplan voor een component moeten altijd uitgevoerd worden, ongeacht of het component al bestaat. Hierbij doel ik op de behoefteanalyse, ontwerpfase en testfase. Het kan immers zijn dat het component toch niet voldoet aan de vernieuwde eisen, het niet geheel aansluit op de nieuwe componenten etc. Elke engineer behoort dit te weten, waardoor ik deze inschattingsfout deze stakeholders zwaar aanreken. Er bestaan foutnormen voor ICT in dit toepassingsgebied. Dit is niet voor niets: één kleine fout en dit leidt tot economische schade en/of menselijk leed.

ESA
De ESA heeft meteen na de rampzalige vlucht een grootschalig onderzoek laten uitvoeren om de oorzaken te achterhalen en eventuele problemen bloot te leggen. De genoemde aanbevelingen zijn allemaal overgenomen en geïmplementeerd in het ontwikkelproces. Je kunt dus stellen dat ESA een adequaat gereageerd heeft op de situatie. In plaats van een intern, geheim onderzoek en met schuldvraag naar anderen te wijzen is er gekozen voor een openlijk onderzoek waarin het boetekleed werd aangetrokken. Op zich waren ze hier natuurlijk ook wel enigszins door gedwongen aangezien de geloofwaardigheid van de hele Europese ruimtevaart op het spel stond.
Wanneer men kijkt naar de oorzaken, kan geconstateerd worden dat ESA geen afdoende risicoanalyse gemaakt heeft voor het gehele ontwikkeltraject. Het is een eerste klas blunder dat niet alle functionaliteit getest hoefde te worden! Dit valt naast de engineers ook de leiding van ESA aan te rekenen. Ruimtevaart staat en valt met de politieke wil van de lidstaten en burgers. Veel mensen zijn van mening dat het onzin is om zoveel geld te stoppen in een dergelijke onderneming. Wanneer deze fouten optreden en er zoveel geld mee gemoeid is, verdwijnt de geloofwaardigheid van de hele onderneming. Dit is ontzettend belangrijk voor een politieke onderneming.

Lidstaten ESA
De lidstaten zijn in dit verband de geldschieters voor het ruimtevaartproject. Veel nieuwe ideeën en producten kunnen op deze manier getest worden in de ruimte, daarnaast biedt het natuurlijk een enorme ‘boost’ voor het aanzien van Europa. De lidstaten hebben, schijnbaar, nooit aangedrongen op een inspectie van de kwaliteit van de verschillende ontwikkelprogramma’s. Dit kan ik me goed voorstellen: ESA is verantwoordelijk voor de ontwikkeling en draagt daarom ook alle verantwoordelijkheid. Ik heb daarom niet het idee dat deze groep stakeholders fout zat.

Analyse ethische, professionele richtlijnen

Om een juist moreel oordeel te vellen, is ondermeer noodzakelijk om duidelijk te krijgen aan welke ethische richtlijnen een professional zich moet houden. Deze richtlijnen komen onder andere terug in de 'engineering doodzonden' en de ethische standaarden van het software engineering vakgebied. Deze richtlijnen zijn opgesteld omdat de uitkomst van deze activiteiten ingrijpt op het leven van de 'normale burger'. De professionals hebben daarom een mate van verantwoordelijkheid naar deze burger toe.

Analyse engineering doodzonden

In [4] worden er een aantal richtlijnen gepresenteerd welke te allen tijde voorkomen moeten worden, zeker bij *safety-critical systems.* Het is daarom belangrijk om deze richtlijnen te bespreken om zo een beter oordeel te kunnen geven. Deze richtlijnen zullen niet per definitie garanderen dat een project succesvol zal zijn, maar voorkomt mogelijke problemen en getuigd van een professionele, verantwoordelijke instelling van de engineer.

1. 'used for effect'
   De keuze voor de ontwikkelomgeving moet ingegeven worden door technische argumenten en niet door politieke. Zo gebeurd het vaak dat de keuze afhangt van de voorkeur van iemand in de ondermening, iets wat niet verantwoord is. In het geval van de Ariane 5 ontwikkeling is gekozen voor de ADA programmeertaal [5]. De beargumentatie voor deze taal is niet aanwezig. Deze taal heeft echter geen expliciete foutafhandeling. Dit is nodig om een programma niet te laten crashen en dwingt de programmeur om goed na te denken over fouten welke mogelijk kunnen optreden. Er kan dus gesteld worden dat er niet de juiste keuze gemaakt is door de engineers!. (Ook al kwam deze fout niet voor in de Ariane 4.)

2. 'exaggeration'
   Het ontwikkelen van een *safety-critical* systeem moet niet gebaseerd zijn op één methode. Er moet een scala aan methoden worden gebruikt om nadelen op te heffen. Helaas is niet bekend of dit gedaan is in deze case. Feit is wel dat bepaalde tests niet zijn uitgevoerd. Dan is toch te concluderen dat niet alle mogelijk methoden gebruikt zijn.

3. 'too trusting'
   Het is niet goed om compleet te vertrouwen op de testmethoden en –technieken. Wanneer een test uitwijst dat er geen fouten gevonden zijn, betekend dit niet dat er ook geen fouten inzitten. Dit principe is wel opgevolgd. In de case study is er echter wel een denkfout gemaakt waardoor niet alle mogelijke simulaties zijn getest.

4. 'rule by a few'
   Het is zeer verstandig om externe kennis in te huren. Meer mensen weten meer dan één. In de voorstellen welke uit het onderzoek zijn gekomen, blijkt dat er externe expertise ingehuurd moet gaan worden. Of dit in de ontwikkelfase ook is gebeurd is niet bekend. Er is daarom niet te concluderen of aan dit principe voldaan is.

5. 'transitory'
   De gekozen ontwikkeltechnieken moeten 'de facto' zijn, langere tijd bruikbaar zijn en voordelen brengen in het ontwikkelproces. Dit is zeker het geval in deze situatie: de software van de Ariane 4 kon zelfs hergebruikt worden! Echter, de engineers zijn in de grootst mogelijk valkuil gestapt: Bij het hergebruiken van code moet alles ook weer uitvoerig getest worden.

6. 'additional words'
   Documentatie is zeer belangrijk bij het ontwikkelen en testen van software helemaal. Dit zal in het geval van de Ariane 5 zeker goed voor elkaar zijn geweest, aangezien er tijdens het onderzoek het probleem tot op de regelcode geanalyseerd kon worden.

7. 'meandering'
   Het is belangrijk om in een vroeg stadium te kiezen voor formele specificatie. Dit voorkomt ambiguïteit en dwingt de ontwerpers en behoefteanalisten om overal goed over na te denken. Dit kan uiteraard het missen van de benodigde tests en nieuwe specificaties kunnen voorkomen. Het is niet duidelijk of de engineers zich aan deze regel gehouden hebben.

Geconcludeerd moet worden dat deze zeven doodzonden bijna allemaal zijn overtreden. De engineers hebben dus geen verantwoordelijke houding aangenomen. Er is dus sprake van moreel falen en dus morele verantwoordelijkheid.

Toepassing ethische standaarden software engineering

Bij het ontwikkelen van een *safety-critical system* moet er gestreefd worden naar het acceptabel niveau waarop fouten kunnen leiden tot verwondingen. Dit is de verantwoordelijkheid van de engineers en het management. Om dit te bereiken is het verstandig om richtlijnen van professionele organisaties (IEEE, ACM) te hanteren tijdens een dergelijk traject. In dit gedeelte van het essay worden een aantal van deze *code of ethics* behandeld en gereflecteerd op deze case.

1. *to accept responsibility in making engineering decisions consistent with the safety, health and welfare of the public or the environment.* (IEEE)
   Volgens de IEEE moet iedere engineer constant rekening houden met de consequenties van bepaalde acties, waarbij veiligheid en (sociaal, economisch) welzijn voorop moeten staan. Nu heeft zich in deze case geen mensenleed voorgedaan. Immers, er is een onbemande raket ontwikkeld. Echter, de engineers moeten zich wel verantwoordelijk opstellen naar de Europese lidstaten en burgers. Er wordt namelijk gewerkt met het geld van de lidstaten en ze hebben een morele verantwoordelijkheid om hiermee goed om te gaan. Geconcludeerd moet dan ook worden dat er beslissingen zijn gemaakt, hoe logisch ze misschien lijken, welke tegen deze ethische code ingaan. Ik heb ook geen grond om te denken dat er een andere afweging was gemaakt wanneer het zou gaan om software voor een bemand ruimtevaartuig.

2. *An essential aim of computing professions is to minimize negative consequences of computing systems, including threats to health and safety.* (ACM)
   Uit al het voorgaande blijkt dat er niet alles is gedaan om de software compleet te testen. Gesteld kan dan ook worden dat men zich niet heeft gehouden aan deze regel.

3. *Avoid harm to others.* (ACM)
   Bij elke te maken beslissing moet afgewogen worden of de consequenties schadelijke gevolgen kunnen hebben. Deze schade hoeft niet per se lichamelijk schade te zijn, maar kan ook economische schade zijn.
   Er is duidelijk niet goed nagedacht over de gevolgen van de verschillende ontwerpbeslissingen. Vreemde is verder dat de beslissing niet primair gebaseerd was op een snellere doorlooptijd of minder kosten, maar op een verkeerde inschatting. Misschien is daarom wel te stellen dat de beslissing zeker niet genomen is door personen welke het *Utilitarianism* framework aanhangen of het *Virtue* framework. De beslissing is gemaakt op gevoel: het idee dat oude, succesvolle software zonder problemen is over te zetten in de nieuwe software. Hier blijkt maar weer eens uit dat een beslissing op gevoel vaak fout kan gaan.

Impact grid

Nu de belangrijkste stakeholders zijn geïdentificeerd, is het zaak om de verschillende ethische aspecten te analyseren per stakeholder (waarbij de engineers en ESA de meeste aandacht zullen krijgen). De *impact grid* is een framework waarmee deze analyse uitgevoerd kan worden.
In het conceptueel framework worden een aantal, mogelijke, ethische aspecten benoemd. Een aantal zijn er echter, in deze case, niet (zeer) relevant, zoals *Quality of Life*, *property rights, privacy* en *Equity and access*. Zo wordt *Quality of Life* niet behandeld aangezien de software voor de Ariane 5 geen inbreuk had op de levenskwaliteit van de burgers in Europa. Misschien wel de producten welke de Ariane 5 mee nam in de ruimte om daar getest te worden, maar dit vind ik een te vergezocht verband. De engineers en de ESA hadden niet de taak om een component te ontwikkelen welke de *Quality of Life* zou verbeteren. Evenmin is er sprake van een conflict op het gebied van de *property rights*. De computersystemen werden binnen ESA ontwikkeld, niet door een derde partij. *Privacy* speelt ook geen rol in deze systemen omdat de software geen informatie over mensen bevat. *Equity and access* wordt buiten beschouwing gelaten aangezien de inzet van ICT in deze case geen directe betrekking had op de burgers. De ESA en de engineers hadden geen ethische keuzes te maken om de systemen breed beschikbaar te maken voor het publiek, het is immers bedrijfssoftware.

Nu is uitgelegd welke aspecten niet behandeld worden, kan er begonnen worden met het uitwerken van de impact grid. Dit zal gedaan worden per stakeholder en per aspect.

Engineer – Use of Power
De engineer heeft de mogelijkheid om keuzes te maken in het ontwikkelproces. Dit geeft de engineer *use of power*. De ontwerp- of ontwikkelkeuzes zorgen ervoor dat anderen hier de gevolgen van ondervinden. In deze case zijn dit de ESA en de lidstaten welke door deze foute keuze veel kosten gemaakt hebben en in diskrediet werden gebracht. Geconcludeerd kan dan ook worden dat de engineers zich meer bewust hadden moeten zijn van de verantwoordelijkheid op dit gebied.

ESA – Use of Power
De ESA heeft de verantwoordelijkheid naar de lidstaten toe om dusdanige ruimtevaartuigen te ontwikkelen waardoor er intensieve testprogramma's in de ruimte uitgevoerd kunnen worden. Wanneer er gekeken wordt naar de soort fout welke gemaakt is, en de veranderingen welke toen zijn doorgevoerd, blijkt dat de testmethoden veel te wensen overliet. De ESA had meer aandacht moeten besteden aan de keuze van ontwikkelomgevingen, -procedures, etc.

Lidstaten – Use of Power
Naar mijn idee treffen de lidstaten geen blaam in deze. De verantwoordelijkheid en macht ligt geheel bij de ESA om te zorgen voor kwalitatief goede ruimteveren.

Engineer – Safety
De engineer heeft de verantwoordelijkheid naar 1) zijn opdrachtgever en 2) de Europese burger / lidstaten om veiligheidsproblemen aan te kaarten. Een goede professional zorgt ervoor dat het juiste methoden worden gekozen om een zo groot mogelijke veiligheid te garanderen. Dit is niet gebeurd; Ada, de programmeer-omgeving, was niet toereikend en in mijn ogen is het een doodzonde als je vergeet om oude componenten niet mee te testen in een nieuwe configuratie. Ook in de begin jaren '90 waren er genoeg testmethoden op de markt om dit goed te testen

ESA - Safety
ESA is verantwoordelijk om procedures in te voeren welke niets aan het toeval overlaten om de veiligheid te garanderen. ESA heeft geen, voldoende, veiligheidsanalyse uitgevoerd naar mogelijke problemen. Dit gaat tegen de verantwoordelijkheid in welke ESA heeft ten opzichte van de lidstaten.

Engineer – Risks and reliability
Uiteraard is het onmogelijk om de eisen dat de systemen 100% betrouwbaar moeten zijn. Professionals horen bekend te zijn met de mogelijkheden en onmogelijkheden van de verschillende ontwikkeltechnieken. Zoals hierboven is uitgelegd is dit dus niet gebeurd, waardoor het safety aspect ook geschaad werd. Er had van de engineers een betere risicoanalyse verwacht mogen worden.

ESA – Risks and reliability
Uiteraard ligt de hoofdverantwoordelijkheid bij ESA zelf voor het maken van een onafdoende risicoanalyse.

Engineer – Honesty and deception
Alle engineers moeten ernaar streven dat genomen beslissingen in alle eerlijkheid en openheid genomen worden. In de gelezen literatuur is niets te vinden over mogelijke fouten welke in deze richting wijzen. Maar aangezien het management nooit iets veranderd heeft aan de testprocedures kan er met grote waarschijnlijkheid gezegd worden dat het management nooit is ingelicht.
Na de fout is er echter door een grootschalig onderzoek, compleet in de openbaarheid, eerlijkheid van zaken gegeven over de oorzaken en blunders aan de kant van de engineers. Er is alleen niet te achterhalen of deze eerlijk vanuit de engineers zelf kwam of vanuit het management.

ESA – Honesty and deception
Feit is dat ESA zeer correct heeft gehandeld na de ontploffing. Het grootschalige onderzoek en de openheid die hierbij in acht werd genomen hebben positief gewerkt op het (politieke) vertrouwen van de lidstaten en burgers. Dit aspect is door de ESA dus zeer goed afgehandeld.

Ethisch oordeel

Nu, na het uitvoerig analyseren van de case, kan er een oordeel gegeven worden over de rol van de engineers en van de ESA in de oorzaak van de ontploffing van de Ariane 5 en de reactie van deze stakeholders hierop.
Via de *codes of ethics* is duidelijke geworden dat engineers ethisch verantwoordelijk zijn in de keuze voor ontwikkelomgevingen en procedures, het uitvoeren van goede veiligheid- en risicoanalyses en het in alle openheid verklaren van keuzen. Is deze ramp met de Ariane 5 de engineers dan te verwijten? JA! Er waren al lange tijd betere methoden en technieken beschikbaar en er is geen voldoende aandacht geschonken aan goede analyses. Er is te kort geschoten op de gebieden van het ontwerp, documentatie, testen en reviewen van de soft- en hardware. Met andere woorden: de *code of ethics* is geschonden door deze groep engineers.

Was het handelen van de ESA correct? Dit is iets lastiger te beantwoorden, maar het management had de engineers wel strakker moeten managen. Echter, de primaire verantwoordelijkheid ligt in mijn ogen bij de engineers, niet bij het management.
De reactie van ESA na de ontploffing verdient een pluim. Er zijn geen machtspelletjes gespeeld, geen rechtzaken gevoerd, maar er is gericht gewerkt naar het vinden van oplossingen. Door dit in alle openheid en eerlijkheid te doen kon het vertrouwen van de lidstaten weer teruggewonnen worden.
Na het onderzoek is er maar één conclusie te trekken: de engineers hebben zich niet gehouden aan afspraken en morele wetten en zijn verantwoordelijk voor de schade. Echter, er is niemand ontslagen! Dit geeft een compleet verkeerd signaal af naar de ICT wereld en de lidstaten. Hoewel een rechtzaak veel geld kost, zou de andere keuze op langere termijn een beter gevolg hebben. Er was dan duidelijk gemaakt dat ook software engineers verantwoordelijkheden hebben en niet zomaar kunnen “hobbyen”.

Om het vraagstuk, of de bouwer of de opdrachtgever verantwoordelijk is voor het falen, op een iets abstracter niveau te brengen, wil ik mijn laatste gedachten hieraan wijden. In andere beroepsgebieden is het niet vreemd dat de bouwer verantwoordelijk is voor eventuele constructiefouten. Zo zal een metselaar de schuld krijgen van het instorten van een gebouw. In de ICT is dit jarenlang een taboe geweest, naar mijn idee omdat software engineering lang gezien werd als een kunststukje en geen gestroomlijnde, professionele bezigheid. Dit is nu aan het veranderen, waardoor de verantwoordelijkheid ook verschoven is. Software engineers hebben nu de tools en de kennis om volledig verantwoordelijk te zijn. Het is daarom zaak dat Software engineers zich gaan gedragen als echte engineers: doelbewust en verantwoordelijk.

Literatuur

[1] Collegeaantekeningen ICT & Samenleving 2
[2] http://www.computingcases.org/general_tools/curriculum/impactcs.html
[3] Stake-holder analyses, ICT & Samenleving 2
[4] “The ethics of safety-critical systems”, J. Bowen
[5] Case study Ariane 5
- Officiële onderzoeksrapport ESA,
- http://www.ima.umn.edu/~arnold/disasters/ariane.html
- http://www.cas.mcmaster.ca/~baber/TechnicalReports/Ariane5/Ariane5.htm
- http://www.rvs.uni-bielefeld.de/publications/Reports/ariane.html
- http://www.cas.mcmaster.ca/~baber/Courses/3J03/StudentPresentations/Ariane5Kamyab.pdf