Integriteit van mens en systeem

Daan Laros
9914579
Informatiekunde

Bachelorscriptie

Integriteit van mens en systeem

27 oktober, 2005

Naam:	Daan Laros
Studentnummer:	9914579
Begeleider:	Luca Consoli

Abstract

Veel processen worden tegenwoordig door software gestuurd en dit aantal neemt nog altijd toe. Dit geldt ook voor processen binnen safety-critical systemen. Het is belangrijk dat de mensen en de systemen die deze processen mogelijk maken integer en van hoge kwaliteit zijn. Om dit te waarborgen zijn er ethische codes opgesteld, die aangeven hoe professionele engineers moeten handelen bij het maken van software. Aangezien er een aantal essentiële verschillen bestaat tussen standaard systemen en safety-critical systemen zal de ethische code voor beide systemen naar alle waarschijnlijkheid ook op een aantal punten verschillen. In deze scriptie worden verschillende relevante aspecten voor een ethische code, speciaal gericht op safety-critical systemen bijeengebracht. De belangrijkste aspecten waar rekening mee gehouden moet worden bij de ontwikkeling en ondersteuning hiervan zullen gegroepeerd in een raamwerk weergegeven worden. Wanneer met al deze zaken rekening gehouden is, kan er gesteld worden dat er een ethische code voor safety-critical systemen ontwikkeld is, die een mogelijke meerwaarde heeft voor toekomstige projecten.

Inhoudsopgave

1 Inleiding

Veel apparaten zijn sterk afhankelijk van een computer, of zijn een vorm van een gedistribueerd computer systeem. Computers hebben een centrale en groeiende rol in de commercie, industrie, overheden, gezondheidszorg, educatie, entertainment en de samenleving in zijn geheel.
Een groep van deze apparaten valt onder de safety-critical systemen. Deze term wordt gebruikt voor systemen waarbij het falen van het systeem leidt tot het in gevaar brengen van het leven, een substantieel economisch verlies of het veroorzaken van aanzienlijke schade aan de omgeving. Dit alles kan samengevat worden onder de noemer onacceptabele consequenties [Knight, 2002].
Een safety-critical systeem mag echter wel falen maar dan moet dit wel op een veilige manier gebeuren. Hierin verschilt het ten opzichte van een normaal betrouwbaar systeem. Een dergelijk systeem faalt niet vaak, maar wanneer dit dan toch gebeurt dan is er geen zekerheid wat er zal gebeuren [Marwedel, 2004].
Software engineers hebben dan ook een cruciale rol bij de ontwikkeling van software systemen. Ze leveren een directe bijdrage aan analyse, specificatie, ontwerp, ontwikkeling, certificatie, onderhoud en het testen van software systemen. Dit geeft hen mogelijkheden om goed of kwaad te doen, of om anderen goed of kwaad te laten doen, of anderen te beïnvloeden goed of kwaad te doen. Gezien de mogelijke impact van de gevolgen moeten software engineers zich houden aan ethische richtlijnen, bijvoorbeeld vastgelegd in een ethische code [Gotterbarn, 1997].
De impact van de mogelijke gevolgen maakt de integriteit van mensen en systemen erg belangrijk. Er moet vertrouwd kunnen worden op de mensen en systemen die een rol spelen bij de ontwikkeling en het gebruik van safety-critical systemen. Iedere activiteit met een wetenschappelijke achtergrond vergt een bepaald niveau van verantwoordelijkheid. Het is belangrijk voor diegenen die hierbij betrokken zijn de bijbehorende morele vraagstukken te begrijpen. De wetenschap heeft verschillende technologieën ontwikkeld die heel handig zijn wanneer ze goed gebruikt worden, maar in potentie ook voor veel schade kunnen zorgen. Gezien de consequenties ten gevolge van fouten met betrekking tot safety-critical systemen is het nog vele malen belangrijker dan bij gewone systemen dat er bij de ontwikkeling rekening gehouden wordt met ethische vragen gerelateerd aan software ontwikkeling [Bowen, 2000].
Zoals eerder beschreven bestaan er essentiële verschillen tussen safety-critical en standaard systemen.

Het verschil in ethische benadering en de mate van ethische verantwoordelijkheid met betrekking tot de software engineering tussen een safety-critical en een standaard systeem vormt de onderzoeksvraag van deze scriptie.
Gezien het verschil in karakteristieken tussen beide systemen zullen de ethische codes en de benodigde ondersteuning hiervoor naar alle waarschijnlijkheid ook verschillen vertonen. Om de verschillen tussen beide systemen aan te kunnen geven wordt een analyse gemaakt van een aantal relevante aspecten van een ethische code speciaal opgesteld voor een safety-critical systeem. De methode die hiervoor gebruikt wordt is een literatuurstudie. Gegevens uit verschillende bronnen worden bijeengebracht en gecombineerd tot relevante informatie. Vervolgens wordt deze verkregen informatie weergegeven in een raamwerk, zodat zichtbaar wordt in welke hoedanigheid bepaalde aspecten voorkomen en hoe ze in relatie met elkaar staan.
Het eindproduct is een overzicht van zaken die een bijdrage kunnen leveren aan een versterking van de integriteit van zowel mens als systeem. Om tot dit product te komen wordt de volgende structuur gehanteerd.
In sectie twee worden de geschiedenis en de inhoud van de verschillende ethische benaderingen behandeld. Tevens worden een aantal belangrijke principes die een rol spelen bij het doen en laten van mensen behandeld. Tenslotte wordt er aangegeven hoe ethiek in de praktijk naar voren komt. Dit omvat het verspreiden van een ethische code. De belangrijkste aanverwante activiteiten worden aangegeven.
In sectie drie wordt vervolgens beschreven hoe een dergelijke code evolueert en op welk gebied de veranderingen liggen. Uiteindelijk wordt aangegeven welke aspecten van invloed zijn op de gebruikswaarden van een ethische code. Dit zijn de kernaspecten die bepalen of een code al dan niet bruikbaar is.
Sectie 4 richt zich op safety-critical systemen. Nu de theoretische en toegepaste ethiek behandeld zijn, wordt aangegeven in wat voor hoedanigheid en in welke onderdelen van de samenleving safety-critical systemen een rol spelen. Vervolgens wordt beschreven welke uitdagingen en welke mogelijke oplossingen er bestaan bij de ontwikkeling van safety-critical systemen.
Zoals Gotterbarn in Software Engineering Code of Ethics beschrijft, is de mens een zeer belangrijke schakel in het software ontwikkelingsproces voor safety-critical systemen. De achtergrond en expertise van de betrokken engineers is dan ook een erg belangrijke factor die bijdraagt tot het succes van een project.
In sectie 5 zal een analyse gegeven worden van de belangrijkste factoren, waar een betrokken engineer rekening mee moet houden.
Software kwaliteit heeft grote invloed op de betrouwbaarheid van een systeem, en is daarmee dan ook een belangrijk onderwerp met betrekking tot een safety-critical systeem.

Sectie 6 zal hierop ingaan.
In sectie 7 wordt tenslotte uit de voorgaande secties verkregen informatie aan elkaar gerelateerd en gegroepeerd. Op deze manier ontstaat er een raamwerk wat een overzicht biedt van de belangrijkste aspecten die komen kijken bij de verhoging van de integriteit van mens en systeem.

2 Ethiek

Deze sectie behandelt de geschiedenis en de inhoud van de belangrijkste ethische benaderingen. Tevens wordt een aantal belangrijke principes die een rol spelen bij het doen en laten van mensen behandeld. Tenslotte wordt er aangegeven hoe ethiek in de praktijk naar voren komt. Dit omvat onder andere het verspreiden van een ethische code. De belangrijkste aanverwante activiteiten worden beschreven.

2.1 Korte geschiedenis

De bakermat van de westerse filosofie, de ethiek en het wetenschappelijk denken kan teruggeleid worden naar de Grieken. In 1892 werd wetenschap aangeduid met de volgende beschrijving: "denken over de wereld op de Griekse manier" [Burnet, 1892].
Het oude Griekenland bood een katalysator voor de initiële toename van kennis. Socrates (469-399 B.C.), Plato (429-347 B.C.) en Aristoteles (284-322 B.C.) hebben filosofisch onderzoek op een zodanige manier ontwikkeld die ver boven het niveau van die tijd stond. Dit was mogelijk door de Griekse manier van leven, deze gaf tijd om na te denken over de meer abstracte vraagstukken. De invloed hiervan is nog steeds zichtbaar. Binnen filosofische kringen worden de ideeën nog steeds actief bediscussieerd.
Een van de producten van het denken uit die tijd zijn de *Nicomachean Ethics*. De Griekse filosoof Aristoteles is de bedenker van deze ethische benadering die een diep en blijvend effect op het westerse filosofische denken heeft sinds het ontstaan ervan in de 4e eeuw B.C. Aristoteles richt zich in dit werk op het object van het leven, het *"goed"* doen. Dit wordt volgens hem bereikt door het uitoefenen van bepaalde "deugden", zoals gerechtigheid en intellectuele activiteit. Deze ethiek is een van de belangrijkste instituten van deugd-ethiek [Hursthouse, 2003].
Aristoteles benadrukt in zijn werk het belang van context bij ethisch gedrag en de vaardigheid van een deugdelijk persoon om de beste actie te herkennen. Daarnaast geloofde hij dat geluk het uiteindelijke doel van het leven is en dat individu's zoektocht naar geluk, leidt tot deugdelijk gedrag. Een deugdelijke persoon moet zijn keuzes dus maken op basis van de context en het uiteindelijke doel. Hij zal altijd die actie moeten kiezen die het best bijdraagt tot geluk. Ieder individu dat rationeel nadenkt zal overwegingen maken op basis van context en consequenties van acties. Niet ieder individu zal geluk echter op dezelfde manier interpreteren, waardoor er toch verschillen ontstaan. Een ethische code is een hulpmiddel om meer duidelijkheid te verkrijgen over het begrip geluk. Dit zal behandeld worden in de volgende paragraaf.

Deugd-ethiek is een van de drie hoofdbenaderingen binnen de ethiek. Een andere gangbare benadering is de deontologische ethiek, deze is gebaseerd op richtlijnen en regels. De benadering stelt dat beslissingen enkel of primair gemaakt moeten worden door te kijken naar iemands plichten en de rechten van anderen. Het stelt dat mensen behoren te leven volgens een aantal vastgestelde principes die niet situatie-afhankelijk zijn. Het belangrijkste aspect van deze benadering is het zo duidelijk mogelijk vast leggen van deze principes, zodat deze doelmatig en algemeen zijn.
Daarnaast bestaat er ook nog het utilitarianisme, deze benadering beschouwt de consequenties van acties op het welzijn van de gehele samenleving. Deze benadering wordt echter niet verder behandeld binnen deze scriptie.

2.2 Ethische codes

De ethiek die beschreven wordt in ethische codes maakt deel uit van de deontologische ethiek [Hursthouse, 2003]. Een organisatie zoals de Association for Computing Machinery (ACM) kan haar leden richtlijnen en regels bieden waarnaar gehandeld moet worden. De code laat tevens zien wat er binnen de gemeenschap als geluk beschouwd wordt. Deze informatie kan een persoon weer gebruiken bij het maken van zijn keuzes. De deugdelijkheid en integriteit van een persoon bepalen echter het belang dat hij aan deze regels en richlijnen hecht. Een goede communicatie over het belang van het handelen volgens de ethische code kan iemand beinvloeden. Er moet veel aandacht besteedt worden aan het attent maken van personen op de gevolgen die hun handelen kan hebben. Dit duidelijk opschrijven en duidelijk maken van de code heeft uiteindelijk als doel dat de personen de code internaliseren. De regels worden dan als vanzelf toegepast en maken deel uit van de denkwijze van de personen. Ook zullen personen die zich de code eigen gemaakt hebben, deze verder communiceren met andere personen, waardoor de code goed verspreid wordt.

2.3 Principes

Ethische overwegingen zijn in het algemeen een bron van veel discussie, zo ook op het gebied van de informatica. Er is geen algemene overeenstemming over de beste benadering, maar de maximalisatie van geluk is vaak het primaire doel. Geluk is hier weer te onderscheiden in individueel geluk en algemeen geluk, deze kunnen met elkaar in conflict zijn. Het risico van dood en verwonding zal

algemeen resulteren in een vermindering van geluk maar soms worden risico's zoals oorlog als een benodigde vorm van kwaad beschouwd [Bowen, 2000].
De ontwikkeling van safety-critical systemen moet als doelstelling hebben het vermijden van het verlies van menselijk leven of serieuze verwondingen door de relevante risico's te beperken tot een acceptabel niveau. Dit wordt normaliter beschouwd als de bepalende factor. Het systeem behoort altijd veilig te zijn, zelfs als dit de beschikbaarheid negatief beinvloedt [Bowen, 2000]. Het is de verantwoordelijkheid van het software engineering team en het management van de betrokken organisatie, te verzekeren dat geschikte mechanismen aanwezig zijn en dat deze op de juiste manier gebruikt worden om dit doel te bereiken voor de levensduur van het product. Hierbij spelen individueel en algemeen geluk een rol. Het algemene geluk behoort altijd boven het persoonlijke gesteld te worden.
Het is verstandig om bepaalde richtlijnen te volgen bij het ontwikkelen van een op software gebaseerd artefact. Zeker wanneer er twijfel over de veiligheid bestaat. De meeste professionele organisaties zoals ACM, IEEE, British Computer Society en de Institution of Electrical Engineers bieden richtlijnen voor hun leden, vastgelegd in een ethische code.
De meeste codes komen vanuit een engineering achtergrond en leggen de prioriteit bij het veiligheidsaspect terwijl de gedragscodes vanuit een informatica achtergrond de nadruk wat breder leggen. Veiligheid blijft nog altijd een belangrijke factor maar andere verliezen worden ook in beschouwing genomen [Bowen, 2000].
In het Verenigd Koninkrijk wordt gebruik gemaakt van richtlijnen die specifiek opgesteld zijn voor engineers en managers die werken aan safety-critical systemen. Deze codes zijn bedoeld voor het gebruik door professionele engineering instituten zoals IEEE. De code vormt een raamwerk, dat door de organisaties verder uitgewerkt kan worden, zodat deze specifiek afgestemd wordt op de desbetreffende sector [Thomas, 1996].
Het is vanzelfsprekend dat een professionele engineer op een rationele manier zijn keuzes moet maken wat betreft de ontwikkeling van een safety-critical systeem. Helaas is de ontwikkeling soms afhankelijk van persoonlijke, vaak ongemotiveerde, voorkeuren van het betrokken personeel. Dit geldt vooral voor personeel met een management-rol. Regels en/of richtlijnen zouden hier een oplossing voor kunnen bieden, door verantwoording van keuzes verplicht te stellen.
Daarnaast kan de door Aristoteles gemaakte verdeling in het rationele denken een handig hulpmiddel zijn voor het analyseren van ethische eisen en richtlijnen. In de wetenschap zijn de theorie en de praktijk twee belangrijk aspecten, die elkaar kunnen bevestigen en/of ondersteunen. Zonder een sterke theoretische

basis kan een praktische applicatie gaan zweven. En zonder een praktische applicatie zijn theoretische ideeën waardeloos. Het onsamenhangende gebruik van praktisch en theoretisch werk is onnatuurlijk en kan negatieve gevolgen hebben. Veel van het praktische werk in de informatica, zowel in software als in hardware engineering, is onjuist en klungelig in elkaar gezet omdat de mensen die het gemaakt hebben geen duidelijk beeld hebben van de fundamentele principes die aan hun werk ten grondslag liggen. Het meest abstracte mathematische en theoretische werk is vaak steriel omdat het geen relatie heeft met echte informatica [Bowen, 2000].
Het is dan ook wenselijk er voor te zorgen dat de scheiding tussen theorie en praktijk geminimaliseerd wordt. Dit is van levensgroot belang op het gebied van safety critical systems. Een goede theoretische en wiskundige onderbouwing is namelijk essentieel om een maximaal begrip van het systeem dat ontwikkeld wordt te verkrijgen. Wanneer dit begrip er niet is kunnen serieuze problemen eenvoudig optreden en kan een oplossing hiervoor erg moeilijk gevonden worden [Bowen, 2000].

2.4 Ethiek in de praktijk

Ethiek komt in de praktijk voor in verschillende vormen. Zo is er natuurlijk het deugd-gedeelte wat deel uit maakt van ieder individu dat rationeel denkt, daarnaast zijn er dan ook de regels en richtlijnen die terugkomen in de ethische code. Een ethische code is op verschillende manieren van belang voor een professionele groepering. Wanneer er tragische gebeurtenissen plaatsvinden, wordt de aandacht gericht op de engineers van het systeem. Er wordt vastgesteld of er fouten zijn gemaakt wat betreft competentie en of beoordeling van de verantwoordelijken. Er wordt nagegaan of de ethische code is overtreden. En of het strikt voldoen aan de code de gebeurtenis zou hebben kunnen voorkomen [Davis, 1991].
Het opstellen van een adequate ethische code is geen gemakkelijke taak. De basiscomponenten van een professionele code zijn vaak eenvoudig te begrijpen, de specifieke details echter vergen enige aanpassing om aan de wensen van een bepaald beroep te voldoen. Deze aanpassingen zullen zorgvuldig moeten plaatsvinden zodat er geen informatie verloren gaat. De complexiteit van dit proces is waarschijnlijk de reden waarom de codes zo weinig herzien worden [Rosenberg, 1998].
Dit heeft als gevolg dat de professionele gemeenschap een actieve en veelomvattende ondersteuning moet aanbieden aan haar leden. Deze moeten hier voor al hun vragen met betrekking tot de ethische code terecht kunnen.

Nieuwe ontwikkelingen kunnen immers nieuwe situaties veroorzaken, waarbij het niet duidelijk is wat de juiste aanpak is. Als onderdeel van het zelfbewustzijn proces, is het schrijven en verspreiden van een ethische code een noodzakelijk en cruciaal aspect. Het bestaan van de code zorgt ervoor dat professionals ethisch handelen, of de door hun zelf opgestelde goede intenties overtreden. Ze worden geacht ethisch en competent te werken en de professionele gemeenschap wordt geacht dit ethisch gedrag te ondersteunen en actie te ondernemen wanneer er van de code afgeweken wordt [Rosenberg, 1998].
Het wordt aangegeven dat de ethische code slechts een gedeelte is van het aanbod van de gemeenschap aan hun leden en de samenleving. Of er daadwerkelijk meer aangeboden wordt hangt van de gemeenschap af.
Het is belangrijk dat een gemeenschap zichzelf wijdt aan het initialiseren en ondersteunen van bijkomstige activiteiten, zoals ethische comités en veldonderzoek, het aankaarten van zaken, het bevorderen van discussies en de behandeling van ethisch gedrag als een belangrijk aspect van de activiteiten van de leden. Met het oog hierop onderscheidt Rosenberg de volgende activiteiten, die toegepast zouden moeten worden door ethische comités binnen professionele gemeenschappen.

- Er moet verzekerd zijn dat leden volledig op de hoogte zijn van de code en dat ze zelf verantwoordelijk zijn voor een basisbegrip van de bedoelingen.
- De verschillende manieren waarop de code onder de aandacht gebracht kan worden bij nieuwe en bestaande leden moet bepleit worden en er moet verantwoordelijkheid genomen worden voor de implementatie ervan.
- Aan de leden moet gepropageerd worden dat het comité beschikbaar is voor consult en advies. Daarbij moeten richtlijnen gesteld worden door de gemeenschap met betrekking tot de legitieme activiteiten betreffende deze zaak.
- De leden moeten jaarlijks via een grote publicatie over de zaken die ter sprake zijn gekomen gerapporteerd worden.
- Er moeten reguliere bijeenkomsten gehouden worden, om te discussiëren over zaken en op grote schaal verslag uit te brengen aan de leden.
- Leden moeten aangemoedigd worden om studenten en faculteiten te informeren over de activiteiten en de mogelijkheden van curriculum ontwikkeling in het betreffende vakgebied.
- Het comité moet dienen als de bewaarplaats van zaken met betrekking tot ethiek en een erkende groep van adviseurs en consultants samenbrengen, wiens expertise gebruikt kan worden voor verschillende doeleinden.

- Aanpassingen onoverkomelijk door het dynamische karakter van de meeste disciplines. Er moeten dan ook aanbevelingen gedaan worden aan de bestuurder van de organisatie voor aanpassingen aan de code, zodat deze adequaat blijft.
- Het comité moet het publieke aanspreekpunt zijn van de organisatie voor vragen over ethiek en professionalisme die voor kunnen komen in de toekomst.

Tegenwoordig is het voor gemeenschappen eenvoudig om hun activiteiten te communiceren via het web. Ze zijn dan ook verplicht de leden goed te informeren en de samenleving te verzekeren dat hun ethische code een belangrijk document is [Rosenberg, 1998]. De samenleving moet vertrouwen hebben in de mensen die werken volgens de code. Dit aspect wordt genoemd omdat het wel degelijk van belang is, het is echter niet het onderwerp van deze scriptie en zal daarom niet verder uitgewerkt worden. Verder moet het duidelijk zijn dat de leden verdedigd worden wanneer ze vals beschuldigd worden tijdens het uitvoeren van hun werk.
De leden moeten gedwongen worden te handelen conform de principes. Tegen onethisch gedrag moet zo snel mogelijke actie ondernomen worden en leden die niet conform de ethische code werken zullen bestraft moeten worden. Een professionele gemeenschap kan niks minder doen [Rosenberg, 1998].

3 Ethische code

Het opstellen van een ethische code is geen gemakkelijke klus, gezien de grote hoeveelheid aspecten waar rekening mee gehouden moet worden. In deze sectie wordt beschreven hoe een code evolueert en op welk gebied de veranderingen liggen. Uiteindelijk wordt aangegeven welke aspecten van invloed zijn op de gebruikswaarden van een ethisch code. Dit zijn de kernaspecten die bepalen of een code al dan niet bruikbaar is.

Als onderdeel van de ontwikkeling van de nieuwe ethische code van de ACM en professioneel gedrag eind vorige eeuw, is er een analyse gedaan om de ethische standaarden te achterhalen van computergebruikers in die tijd.
Door een vergelijk te maken tussen de bestaande codes van de ACM, het Institute of Electrical and Electronic Engineers (IEEE), de Data Processing Managers Association (DPMA) en het Institute for Certification of Computer Professionals (ICCP), kwamen er een aantal kernthema's naar voren. De kern van ethisch gedrag zoals deze geaccepteerd wordt door computer engineers is de volgende [Martin, 1990].

- Personal integrity / claim of competence;
- Personal accountability for work;
- Responsibility to employer / client;
- Responsibility to profession;
- Confidentiality of information / privacy;
- Conflict of interest;
- Dignity / worth of people;
- Public safety, health en welfare;
- Participation in professional societies;
- Increase public knowledge about technology.

Het is goed om te zien dat in de verschillende ethische codes de nadruk ligt op de relatie tussen de computergebruiker met andere mensen in plaats van machines. Dit legde de focus van ethisch gedrag op het ethische, of juiste omgang met mensen, in plaats van machine gecentreerd [Martin, 1998].
De kernaspecten leggen in veel gevallen de verantwoordelijkheid bij het individu. Dit geeft dan ook weer aan hoe belangrijk deugd-ethiek en een goede communicatie van de ethische code is. Mensen moeten het belang van dergelijke aspecten goed in kunnen schatten en het moet duidelijk zijn met welk doel bepaalde beslissingen gemaakt moeten worden.
Aangezien de focus binnen de verschillende codes op dezelfde aspecten ligt, kan een organisatie zich enkel onderscheiden door nieuwe aspecten onderdeel te

maken van de ethische code en daarmee een meer aan de omstandigheden aangepaste code te ontwikkelen. Deze uitdaging nam de ACM aan.
Tijdens de ontwikkeling van de nieuwe ethische code van de ACM werd er gewerkt aan de ontwikkeling van een internationale ethische code voor informatie-technologie-professionals door de International Federation of Information Processing (IFIP). Dit heeft uiteindelijk geen product opgeleverd.
Het raamwerk, waarbinnen de code moest komen, heeft echter wel bijgedragen aan de ontwikkeling van de nieuwe ethische code. De reden van falen had betrekking op de gebieden die betrokken werden in het raamwerk. Het kwam in gebieden zoals internationaal recht en culturele normen en waarden, wat veel verder ging dan de op dat moment geaccepteerde ethiek [Martin, 1998].
Het raamwerk onderscheidde acht nieuwe thema's. Deze thema's omvatten specifieke uitspraken met betrekking tot sociale verantwoordelijkheid, het vaststellen van standaarden, de nadruk op kwaliteit van leven, gelijkheid, bescherming van intellectueel eigendom, consequenties van netwerken, grond mensenrechten en de rechten van gebruikers. Enkele van deze nieuwe ideeën zijn uiteindelijk onderdeel geworden van de ethische code opgesteld door het ACM. De nieuwe ACM code is daarmee breder dan zijn voorganger en beter aangepast aan de huidige omstandigheden [Martin, 1998].
Om een goed beeld te vormen van de verschillen tussen beide codes, zal een beeld geschetst worden van beide.
De oude code bestond uit vijf brede categorieën met daaronder vijftien ethische overwegingen. Deze werden aangeraden maar waren niet bindend. Voor iedere overweging waren er dan weer een of meerdere disciplinaire regels. Deze beschreven heel specifieke acties en werden bindend beschouwd voor het lidmaatschap van de ACM. De ethische overwegingen en disciplinaire regels werden gecommuniceerd met alle leden van de ACM, zonder te kijken naar de mate van professionele verantwoordelijkheid. De nadruk bij de oude code lag op de verantwoordelijkheid van het individu op het competent uitvoeren van de toegewezen taken door de werkgever. De nieuwe code omvat alle zaken van de oude code maar legt de nadruk meer bij de algemene sociale verantwoordelijkheid van de professional [Martin, 1998]. De nieuwe code is sterker gericht op de kernpunten die naar voren kwamen uit het vergelijk van de verschillende codes. Deze kernpunten die veelal te maken hadden met professionele verantwoordelijkheid zijn samengevoegd met punten van algemene sociale verantwoordelijkheid.
De nieuwe verdeling bestaat uit vier secties[1] bestaande uit 24 imperatieven. Deze imperatieven zijn geformuleerd als verklaringen van persoonlijke

[1] http://www.acm.org/constitution/code.html

verantwoordelijkheid. In combinatie hiermee is er een set van richtlijnen die het individu sturen met betrekking tot de interpretatie en het gebruik van de opgestelde code bij het nemen van beslissingen [Martin, 1998].
De eerste sectie bestaat uit algemene morele imperatieven die de persoon sturen om professioneel gedrag toe te passen. De tweede sectie behandelt meer specifieke professionele verantwoordelijkheden die geïnterpreteerd moeten worden aan de hand van de IT professie. De derde sectie voegt een nieuwe dimensie toe aan de gedragscode, het presenteert de imperatieven die een rol spelen bij organisationeel leiderschap. De brede verklaringen worden gerelateerd aan algemene sociale verantwoordelijkheid, kwaliteit van leven, waardigheid en gebruikersrechten, en ze behandelen veel van de zaken voorgesteld in het IFIP raamwerk. De vierde sectie behandelt overeenkomstigheid. De verklaring stelt dat overtredingen inconsistent zijn met het lidmaatschap in de ACM [Martin, 1998].
Verder laat een thematische analyse van de nieuwe code zien dat er veel dezelfde ethische vraagstukken behandeld worden als in de oude code. De algemene vorm van de nieuwe code verschilt echter substantieel van de oude. De oude code geeft gedetailleerd aan wat wel en niet kan en organiseert dit onder de vijf categorieën. De nieuwe code omvat een bredere scope, de secties zijn georganiseerd op basis van toenemende niveaus van verantwoordelijkheid als professional. De eerste sectie omvat persoonlijke verantwoordelijkheid. De tweede de verantwoordelijkheden van een computer professional met specifieke ethische zaken gerelateerd aan de informatica. En de derde met de verantwoordelijkheden van een manager die geacht wordt rekening te houden met de sociale impact van computer systemen. In sommige aspecten is de nieuwe code minder specifiek dan de oude. Maar het behandelt wel meer onderwerpen. De richtlijnen die aan de nieuwe code toegevoegd zijn bieden tevens meer hulp bij het assisteren van een lid om de imperatieven te interpreteren met betrekking tot de besluitvorming [Martin, 1998].
Algemeen gezien biedt de nieuwe code een sterkere set van beperking en meer richting aan het doen en laten van ethisch professioneel gedrag dan de voorgaande code. Er treedt een duidelijke verschuiving op betreffende de verantwoordelijkheid. De code evolueert en de verantwoordelijkheid wordt meer bij het individu gelegd. Een individu wordt geacht niet alleen professionele verantwoordelijkheid te zijn maar ook algemene sociaal verantwoordelijk. Dit leidt er toe dat er een combinatie gemaakt kan worden tussen deugd en deontologische ethiek. Het individu moet nu persoonlijke moraliteit in combinatie met zijn interpretatie van de opgestelde code gebruiken om zo rationeel mogelijk beslissingen te nemen.

3.1 Gebruikswaarde van een code

Een fundamentele vraag die bij iedere ethische code gesteld kan worden is of deze gebruikt kan worden door de software engineer bij het maken van ethische besluiten. Om bruikbaar te zijn bij het maken van dergelijke besluiten moet de ethische code aan de volgende karakteristieken voldoen [Frankel, 1989].

- Het document moet de beoefenaar in staat stellen om de juiste beslissingen te nemen.
- Het moet afschrikwekkend werken tegenover onethisch gedrag.
- Het moet een bron van publieke evaluatie van het beroep zijn en de reputatie van het beroep verbeteren.
- Het moet een bron zijn van professionele socialisatie, het moet beoefenaars helpen om sociaal te worden met betrekking tot hun gedeelde waarden en doelstellingen van hun beroep .
- Het biedt een ondersteunend systeem (moreel, wettelijk, financieel) aan leden die moeilijke ethische beslissingen moeten nemen. Dit systeem zal leden onder andere ondersteunen bij rechtszaken.
- Het biedt een basis voor de berechting van ethische controverses.

De nieuwe ACM code en professioneel gedrag is een voorbeeld van een groeiend besef van sociale verantwoordelijkheid, dat samen is gegaan met de technologische ontwikkeling [Martin, 1998]. De verscheidenheid aan ethische zaken ten gevolge van computertechnologie is groot, waardoor de ACM meer mogelijkheden heeft om met deze toekomstige zaken om te gaan.
Het punt dat de ethische code de beoefenaars moet helpen om sociaal te worden met betrekking tot hun gedeelde waarden en doelstellingen van hun beroep is een voorbeeld van de relatie tussen deugd en deontologische ethiek. De regels en richtlijnen kunnen een individu helpen om zijn deugdelijkheid aan te passen en hiermee vanuit zijn eigen moraliteit de juiste beslissingen te nemen.
Verder wordt het belang van een goed ondersteunend systeem[2] opnieuw aangegeven.
Er wordt echter niks vermeld over de instelling van certificatie standaarden voor computer professionals. In sectie 5 zal hier dieper op ingegaan worden.

[2] Zie sectie 2.2.

4 Safety-critical systemen

Nu de theoretische en toegepaste ethiek behandeld zijn, is het de beurt aan de safety-critical systemen. Er wordt aangegeven in wat voor hoedanigheid en in welke onderdelen van de samenleving dergelijke systemen een rol spelen. Vervolgens wordt beschreven welke uitdagingen bestaan, en welke mogelijke oplossingen er zijn bij de ontwikkeling van safety-critical systemen.

Het aantal systemen dat beschouwd wordt als safety-critical blijft toenemen. De lage hardware kosten, de toename van kwaliteit van hardware en andere technologische ontwikkelingen zorgen ervoor dat er veel nieuwe software systemen gebruikt zullen gaan worden in verschillende domeinen. Bij dit alles is het wel belangrijk om op te merken dat software beslissingen vaak genomen worden op basis van economische factoren in plaats van veiligheidsaspecten [Bowen, 2000]. Dit is natuurlijk niet het juiste uitgangspunt voor een systeem waarbij de veiligheid het belangrijkste aspect is. Het gebruik van al deze systemen brengt dan ook veel nieuwe uitdagingen met zich mee. Aan deze uitdagingen zal voldaan moeten worden om de kwaliteit van producten te garanderen.

4.1 Soorten safety-critical systemen

Het belangrijkste aspect dat bepaalt of een systeem safety-critical is, is zowel vanuit formeel als intuïtief oogpunt de consequentie die het falen van een dergelijk systeem met zich meebrengt. Wanneer het falen van een systeem leidt tot consequenties die beschouwd worden als onacceptabel dan wordt het systeem safety-critical genoemd [Knight, 2002]. In essentie is een systeem dan ook safety-critical wanneer de samenleving ervan afhankelijk is. Traditionele gebieden waar safety-critical systemen voorkomen zijn de gezondheidszorg, de vliegtuigindustrie, kerncentrales en de wapenindustrie. Door het veelzijdige gebruik van computersystemen zijn er tegenwoordig ook veel nieuwe systemen die als safety-critical beschouwd kunnen worden. Het telefoonsysteem achter het alarmnummer is een voorbeeld van een dergelijk systeem. Een verlies, al dan niet tijdelijk, hiervan zal zeker tot een groot aantal doden en gewonden leiden. Ook verkeerscontrole systemen, bank en financiële systemen, elektriciteits distributie en productie-systemen vallen in deze categorie. Het falen van dergelijke systemen leidt tot grote verstoringen van de dagelijkse activiteiten van een groot gedeelte van de samenleving [Knight, 2002].
Bovenstaande voorbeelden van safety-critical systemen omvatten operationele systemen. Software kan echter ook op heel andere manieren gebruikt worden. Zo

zijn er software systemen die gebruikt worden bij het ontwerp en de bouw van andere systemen. Software die de ontwikkeling van andere software ondersteunt, zoals een compiler, is zelf ook safety-critical wanneer het ondersteunde product safety-critical is. Ditzelfde geldt voor computersystemen die de ontwikkeling van niet-computer artefacten ondersteunen, een voorbeeld hiervan is MSC Corporation's NASTRAN. NASTRAN is een systeem waar veel gebruik van gemaakt wordt bij het maken van structurele analyses. Er wordt op een juiste analyse gerekend die gebruikt wordt bij de assistentie van structureel ontwerp automatisering. De analyse wordt door de structurele engineers die hier gebruik van maken geacht te kloppen. Een fout in de analyse zal naar alle waarschijnlijkheid leiden tot een fout in het uiteindelijke product dat een gebouw, een brug of een andere vorm van fysieke architectuur kan zijn [Knight, 2002].
Ook informatiesystemen kunnen safety-critical systemen genoemd worden. De klassering hangt hierbij af van de eigenschappen van de betreffende informatie. De consequenties van het verlies van bepaalde informatie, door onvoldoende of onvolledige security, kunnen enorm zijn. Privé-netwerken gebruikt door financiële instellingen worden gebruikt om geld rond te sturen. Een succesvolle aanval op een dergelijk systeem kan geld of belangrijke informatie zoals credit card nummers of andere persoonlijke gegevens opleveren.
Gezien de verschillende vormen van safety-critical systemen en de verschillende gebieden waarin ze voorkomen, wordt het meteen duidelijk waarom het zo belangrijk is dat er zoveel aandacht besteedt wordt aan dergelijke systemen. De hele samenleving is afhankelijk van dergelijke systemen.

4.2 Uitdagingen

Software engineers moeten met verschillende aspecten rekening houden bij het creëren van safety-critical systemen. In deze paragraaf volgt eerst een aantal aspecten dat primair te maken heeft met de specificatie van het te ontwikkelen systeem. Vervolgens wordt een aantal aspecten behandeld dat te maken heeft met de verificatie en validatie van het systeem, uiteindelijk gevolgd door security aspecten. Het is echter niet de techniek die het hardst aan verbetering toe is, maar de betrokkenen. De aandachtspunten die van belang zijn voor mensen die werken aan een safety-critical systeem worden dan ook genoemd.

Het aantal safety-critical systemen dat met elkaar communiceert in een enkele applicatie zorgt ervoor dat er resources gedeeld moeten worden tussen systemen. Dit elimineert het architecturale element, fysieke separatie, dat vertrouwen moet geven voor een correcte werking. Het falen van een systeem

mag immers geen invloed hebben op het correct werken van een ander systeem. Dit gaat verloren wanneer er veel interactie is tussen verschillende systemen, ze worden dan afhankelijk van elkaar [Knight, 2002]. Dit is zo wanneer meerdere functies gesitueerd zijn op een platform zodat de constructie eenvoudiger is. Er dienen dan ook technieken ontwikkeld te worden, die een hoge zekerheid bieden van non-interventie.
Het falen in de interactie tussen software en systeem engineering veroorzaakt een groot gedeelte van de fouten. Het is van essentieel belang dat veelomvattende benaderingen voor totaal systeem-modellering ontwikkeld worden zodat de eigenschappen van volledige systemen geanalyseerd kunnen worden. Een dergelijke benadering moet software op de juiste manier afstemmen en moet betrouwbare modellen van kritieke software karakteristieken bieden. Ook moet het aspect van non-interventie meegenomen worden [Knight, 2002].
Fouten in de software-specificatie spelen een rol in vele vormen van falen. Vele vormen van de specificatie worden niet ondersteund door de huidige technieken. En waar deze technieken al bestaan, daar ontbreekt de integratie om een volledige specificatie-analyse te maken [Knight, 2002].
Bovenstaande aspecten zijn alle gerelateerd aan de specificatie. Wanneer voor het ontwerp een duidelijk plan gemaakt is, kunnen deze problemen voorkomen worden. Voordat er aan de ontwikkeling van een systeem begonnen kan worden, is het dan ook noodzakelijk dat er een goede uitgewerkte specificatie is, waarin een oplossing gevonden is voor dergelijke problemen.

Verificatie door middel van testen is onmogelijk voor systemen die werken op een hoog afhankelijkheids niveau. Formele verificatie en model checking worden dan ook vaak gebruikt bij dergelijke systemen. Deze zijn echter gelimiteerd in hun mogelijkheden [Knight, 2002].
Software verificatie en validatie kunnen extra betrouwbaarheid geven maar metingen van de effectiviteit van dergelijke benaderingen zijn moeilijk te verkrijgen. Dit betekent dat het volledig steunen op dergelijke software een gewaagde keuze is. Ondanks dit alles wordt er vaak gedacht dat software betrouwbaarder is dan hardware omdat het niet veroudert op eenzelfde manier als hardware doet. In tegenstelling hiertoe treden fouten echter op een veel toevalligere en onvoorspelbare manier op. Software is een digitaal artefact. Dit betekent dat technieken zoals inter- en extrapolatie, zoals deze gebruikt worden door veel hardware engineers in berekeningen, niet gelden. Het veranderen van een enkele bit in een computerprogramma kan een onvoorspelbaar effect hebben op de operatie, maar voor hetzelfde geld blijft het onzichtbaar. De complexiteit van de meeste software ligt zo hoog, dat het extreem moeilijk is om het gedrag met redelijke zekerheid te kunnen voorspellen. Gegeven al deze problemen van

software zal er bij het bepalen voor het gebruik van software in safety-critical systemen goed nagedacht moeten worden over de software keus. De keus moet immers gemaakt worden met als doel het verkrijgen van een zo hoog mogelijke zekerheid dat het systeem niet faalt. Het is dan ook aan te raden back-up systemen te gebruiken die erg eenvoudig werken [Knight, 2002].
Hoge prestatie, snelle en veelomvattende benaderingen tot verificatie zijn dan ook essentieel wanneer de safety-critical systemen betrouwbaar zijn.
Het is verstandig om software zoveel mogelijk te verifieren en valideren, zodat er met redelijke zekerheid iets over de betrouwbaarheid gezegd kan worden. Dit kan alleen gebeuren indien de software niet te complex is. Om dit te kunnen verwezenlijken zal de specificatie hierop aangepast moeten worden. Tevens is het van belang dat de juiste verificatie- en validatie-technieken gebruikt worden zodat er een nauwkeurige analyse gemaakt kan worden.
Bovenstaande aspecten hangen samen met de huidige beperkingen van software engineering. Software engineering van safety-critical systemen is een complexe taak die betrekking heeft op verschillende technische gebieden. Software is een kerncomponent van elke safety-critical systeem. Er zijn echter veel te weinig ontwikkelaars in andere disciplines die begrijpen wat software wel en niet kan. Mensen moeten tijdens hun opleiding al op de hoogte gesteld worden over de aspecten die komen kijken bij safety-critical systemen [Knight, 2002]. Hierdoor kan veel tijd bespaard worden, door het verminderen van wensen die niet uitvoerbaar zijn.
Daarnaast is security een onderwerp dat steeds belangrijker wordt voor safety-critical systemen. Het moet veelomvattend beschouwd worden wanneer safety-critical systemen succesvol moeten opereren. De uitdaging ligt hier vooral op het gebied van software engineering en in mindere mate in de security technologie. De meerderheid van de security problemen treedt op in informatie systemen die door middel van een netwerk verbonden zijn. Fouten in software maken deze kwetsbaar voor aanvallen. De achtergrond achter aanvallen is vaak wel duidelijk, het ervoor zorgen dat deze aanvallen niet uitgevoerd kunnen worden door het maken van software die niet kwetsbaar is, is een stuk moeilijker.
Om de verschillende problemen het hoofd te kunnen bieden is het van belang dat engineers en managers die een aandeel hebben in een safety-critical systeem rekening houden met een aantal zaken. Het Hazards Forum[3] heeft hiervoor enkele richtlijnen opgesteld.

[3] The Hazards Forum, *Safety-related systems: Guidance for engineers*, 1995

- De gevolgen van hun werk mogen geen onacceptabele risico's voor de veiligheid opleveren. Er moet verzekerd zijn dat alle redelijkerwijs mogelijke zorg om dit te voorkomen gedaan is.
- Maak geen pretenties over werk die niet waar zijn of misleidend zijn, of niet ondersteund worden door de erkende denkwijze binnen het betreffende vakgebied.
- Accepteer persoonlijke verantwoordelijkheid voor al het werk dat gedaan is door uzelf of onder uw verantwoordelijkheid.
- Neem alle mogelijke stappen om competenties te behouden en te ontwikkelen door aandacht te schenken aan nieuwe ontwikkelingen in wetenschap en engineering die relevant zijn voor het vakgebied. Moedig anderen onder uw supervisie aan ditzelfde te doen.
- Geef de beperkingen van anderen aan wanneer ze geloven niet competent te zijn om bepaalde taken uit te voeren en geef dergelijke beperkingen aan wanneer deze aanwijsbaar zijn nadat de taak begonnen is.
- Neem alle redelijkerwijs te verwachten stappen om uw eigen managers en anderen die als plicht hebben zorg te dragen, op de hoogte te stellen van de risico's die u geïdentificeerd hebt. Stel iedereen die professioneel advies negeert of overruled formeel op de hoogte van de daaropvolgende risico's.
- Neem alle redelijkerwijs te verwachten stappen om ervoor te zorgen dat de personen onder uw supervisie of directie competent zijn. Deze personen moeten op de hoogte zijn van hun eigen verantwoordelijkheden en ze moeten persoonlijke verantwoordelijkheid voor het werk dat aan hen gedelegeerd wordt nemen.

Iedereen die verantwoordelijk is voor human resource toewijzing behoort:

- Alle redelijkerwijs te verwachten zorg dragen dat het toegewezen personeel competent zal zijn voor de taken die ze toegewezen zullen krijgen.
- Te verzekeren dat human resources geschikt zijn om de geplande taken uit te kunnen voeren.
- Te verzekeren dat voldoende resources aanwezig zijn om het variatie niveau te bieden geschikt voor de bedoelde samenhang.

Wanneer engineers en managers met bovenstaande aspecten rekening houden kunnen problemen voorkomen, dan wel sneller opgelost worden. Het probleem wordt hiermee ook op de meest toegankelijke plaats aangepakt. Betrokkenen zouden veel problemen kunnen voorkomen door vooraf goed na te denken,

weloverwogen beslissingen te nemen en de verantwoordelijkheid hiervoor te dragen.
Een minder toegankelijk probleem is het volgende. Inspanningen en tijd bij de ontwikkeling van safety-critical systemen zijn zo extreem met de huidige technologie dat het bouwen van toekomstige systemen in veel gevallen niet mogelijk is. Nieuwe software moet oplossingen bieden voor deze tijd en kosten problemen. Er is hier echt een aanzienlijke verbetering nodig. Waarschijnlijk zal er voor dit probleem niet op korte termijn een oplossing gevonden zijn. Dit maakt de eerder beschreven aspecten relevant voor betrokkenen nog veel belangrijker. Op die manier wordt het probleem immers aangepakt op een plaats waar het ook daadwerkelijk tot verbeteringen leidt.

5 Denkstijlen

Zoals al eerder beschreven is de mens een zeer belangrijke schakel in het software ontwikkelingsproces voor safety-critical systemen. De achtergrond en expertise van de betrokken engineers is dan ook een erg belangrijke factor die bijdraagt tot het succes van een project. In deze sectie zal een analyse gegeven worden van de belangrijkste factoren opgesteld door Jonathan Bowen, waar een betrokken engineer rekening mee moet houden.

5.1 Wetenschap of wetenschappelijke kennis

Gezien de sterke relatie tussen theorie en praktijk[4] is het belangrijk dat een praktische applicatie gebaseerd is op een theoretische basis. Dit wordt doorgaans bereikt door basis educatie, uiteindelijk gevolgd door specialistische cursussen en trainingen. Basis educatie kan verkregen worden door het raadplegen van software engineering tekst- en referentieboeken. Deze bevatten gewoonlijk een sectie over safety-critical systemen [Bowen, 2000]. Bowen stelt dat dit gevolgd moet worden door professionele ontwikkeling, wat van cruciaal belang is om bij blijven bij de veranderingen in de betreffende sector. Er zijn verschillende cursussen aanwezig om een Master of Science titel te behalen in safety-critical systems[5]. Een engineer kan op deze manier een certificaat behalen op het gebied van safety-critical systems. Hij toont hiermee aan dat hij competent is voor het betreffende werk. Het volgen van een dergelijke cursus is echter geen verplichting. Om de kwaliteit van de personen die aan safety-critical systemen werken te verhogen, zou het volgen van een dergelijke cursus, inclusief aanvullingen, verplicht gesteld moeten worden. De inhoud van deze cursus zal goed in de gaten gehouden moeten worden, zodat deze in overeenstemming met met de werkelijkheid blijft. Het curriculum van de cursus zal dus dynamisch ingericht moeten worden, zodat het eenvoudig is om veranderingen te implementeren. Engineers zullen ook aanvullende cursussen moeten volgen, indien er belangrijke veranderingen in het vakgebied zijn.

5.2 Kunst of technische vaardigheid

Wanneer de educatieve basis gelegd is, is het van belang dat de nieuw verkregen theoretische kennis omgezet wordt in praktijkervaring. Voor sommige

[4] Zie sectie 2.1

[5] University of York, The, Safety critical systems engineering, system safety engineering: Modular MSc, diploma, certificate, short courses, 2005, The University of York, Heslington, U.K., http://www.cs.york.ac.uk/MSc/SCSE/

vaardigheden, zoals mathematische vaardigheden is dit vaak nog complexer dan het verkrijgen ervan in eerste instantie. Een van de moeilijkste aspecten is het met voldoende nauwgezetheid leren modelleren van de realiteit. Dit vormt een heel belangrijk aspect bij computer engineering, aangezien dit het verdere verloop van het project bepaalt. De abstractie die hiervoor nodig is, is een vaardigheid die alleen verkregen kan worden door te werken in de praktijk. Helaas moeten veel programmeurs, wanneer ze het systeem als een geheel beschouwen, hun natuurlijke wil om te veel details bij de implementatie te betrekken afleren. Alleen de relevante aspecten moeten aanwezig zijn bij een bepaald abstractieniveau [Bowen, 2000]. Om dit probleem te verminderen, is het van belang dat engineers die nieuw in het vak zijn, goed begeleid worden door senior-engineers die een rijke verzameling aan vaardigheden hebben. Het is natuurlijk wel van belang dat deze senior-engineers over voldoende communicatieve en sociale vaardigheden beschikken, zodat zij hun kennis ook over kunnen dragen aan de nieuwelingen. Op deze manier wordt er gebruik gemaakt van de lerende organisatie.

5.3 Beleid of praktische wijsheid

De gevolgen van fouten bij safety-critical systemen kunnen een grote impact hebben. Daarom is het belangrijk dat nieuwe technieken geleidelijk gebracht en voorzichtig gebruikt worden in safety-critical systemen. Het is belangrijk dat deze voorzichtigheid geen negatieve invloed op de ontwikkeling heeft. Om toch nieuwe benaderingen te kunnen testen is het aan te raden om deze te gebruiken op non-safety-critical systemen om vertrouwen te krijgen, ook wanneer ze theoretisch nog zo veelbelovend lijken, en volledig begrepen worden door de ontwikkelaars. Wanneer er voldoende ervaring is verkregen, en de baten zijn geanalyseerd, dan kan de techniek aangenomen worden voor het gebruik in safety-critical gebieden [Bowen, 2000]. Om dit traject goed te laten verlopen is het belangrijk dat de voorschrijvende aanbevelingen in standaarden vaak geüpdate worden, zodat de gegevens zo accuraat mogelijk blijven. Hierdoor is het voor alle partijen duidelijk welke technieken onderzocht worden, en welke technieken geschikt zijn bevonden voor het gebruik in safety-critical systemen. Tevens zal het curriculum van de cursussen herzien moeten worden, zodat ook de nieuwste technieken behandeld worden. De certificatie van de engineers moet rekening houden met verschillende kennisniveaus. Een engineer die een aanvullende cursus met nieuwe informatie succesvol heeft afgesloten zal een andere certificatie moeten hebben dan iemand die dit nog niet heeft gedaan.

Iedere engineer heeft zijn beperkingen, het is belangrijk dat inviduen hun eigen beperkingen en die van anderen herkennen. Het is belangrijk dat individuen werken binnen hun mogelijkheden en indien dit niet zo is dat zij hier op gewezen worden door collega's of werkgevers. Hoe competent een persoon ook is, er zijn altijd taken die niet met rationaliteit opgelost kunnen worden. Wanneer de beperkingen op de ontwikkeling van een safety-critical systeem onmogelijk behaald kunnen worden, dan behoren er mechanismen te zijn die het mogelijk maken dat dit duidelijk gemaakt kan worden door de gehele organisatie [Bowen, 2000].
Een duidelijk voorbeeld waar deze mechanismen niet effectief waren is de Therac-25 radiatie therapie-machine. Hier traden verschillende fatale radiatie overdoses op, door een obscure hapering van het materiaal. De hapering was het gevolg van de software. De software op zichzelf was echter niet onveilig, de combinatie met de hardware en de omgeving creëerde echter een volledig systeem wat niet veilig was [Leveson, 1993].

5.4 Intelligentie of intuïtie

De kwalificaties van het personeel hebben veel invloed op het algemene kwaliteitsniveau. Personeel met de juiste kwalificaties behoort dan ook werkzaam te zijn bij de ontwikkeling van safety-critical systemen. De betrokken engineers moeten capabel zijn om de benodigde kennis gereedschappen te gebruiken en een goed begrip te hebben van de benodigde operaties van computergebaseerde systemen [Bowen, 2000]. De certificatie kan dit selectieproces vereenvoudigen. Het proces kan zich op deze manier meer concentreren op bijkomstige vaardigheden (sociale, communicatieve).
Specialistische technieken, zoals het gebruik van wiskunde en abstractie zijn belangrijk bij het construeren van systemen met de hoogste integriteit. Formele specificatie helpt het voorkomen van fouten, voor lage kosten of zelfs een beperking van de totale kosten. Formele verificatie, hoewel dit zeer duur is, kan het aantal mogelijke fouten nog verder verminderen en kan kosten effectief blijken wanneer de prijs van falen extreem hoog is. Hierbij is het van belang dat de juiste verificatie en validatie-technieken gebruikt worden zodat er een nauwkeurige analyse gemaakt kan worden [Bowen, 2000].
Om volledig vertrouwen in het systeem te hebben moet gebruik worden gemaakt van machine ondersteuning. Dit kan gebruikt worden om het juistheids bewijs van de software implementatie met betrekking tot de formele specificatie te mechaniseren. Mechanisatie van bewijzen kan dit proces aanzienlijk versnellen er zitten echter ook een aantal negatieve aspecten aan. Voor traditionele

wiskundigen, is het vormen van een bewijs een sociaal proces dat jaren, decennia of zelfs eeuwen kost om algemeen geaccepteerd te worden. De automatisering van bewijzen wordt dan ook met argusogen bekeken door deze groep. Een kritisch aspect van geautomatiseerde bewijzen, is het ontbreken van de mogelijkheid tot inspectie. Hierdoor kan er geen begrip verkregen worden over de validiteit van een bewijs en daarmee is er geen ondersteuning voor de juistheid ervan. Dit komt omdat een geautomatiseerd bewijs kan bestaan uit miljoenen, simpele, stappen. Deze zijn echter bijzonder moeilijk te volgen tot het resultaat. Bovendien moeten de meeste bewijzen die uitgevoerd worden binnen de software engineering industrie een stuk sneller uitgevoerd en geaccepteerd worden dan traditionele wiskundige bewijzen. Het gaat hier vaak om weken of maanden in tegenstelling tot jaren en decennia. Echter zijn ze wel een stuk oppervlakkiger dan de meeste bewijzen waarin professionele wiskundigen geïnteresseerd zijn [Bowen, 2000]. Dit alles zorgt er wel voor dat intuïtie, waarom een programma correct is aan zijn specificatie, zo moeilijk te verkrijgen is in de meeste gevallen. Het geautomatiseerde bewijs blijft van hoge complexiteit, en is niet eenvoudig te doorgronden. Dit probleem is ook niet eenvoudig op te lossen er zal dus nog een hoop onderzoek op dit gebied plaats moeten vinden. Samenwerkingsverbanden zoals de European Provably Correct Systems (PROCOS) projecten proberen de formele technieken die te maken hebben met de formele requirements analyse te onderzoeken. Dit wordt gedaan door te kijken naar de verschillen wat betreft ontwerp-niveau, programmeren, compileren, en zelfs tot in de transformaties die de hardware gebruikt gebaseerd op algebraïsche wetten. Het samenvoegen van theorieën voor deze verschillende niveaus is nodig om de consistentie te waarborgen. Het is echter moeilijk om de compatibiliteit van de gebruikte modellen te garanderen zonder de flexibiliteit te beperken. Een mogelijkheid om duidelijke begrijpbare geautomatiseerde bewijzen te maken zal er voorlopig nog niet zijn. Gezien het vele onderzoek is het ontstaan hiervan in de toekomst wel degelijk een mogelijkheid.
Zoals ook beschreven in paragraaf 4.2 vormt complexiteit een van de grootste problemen. Het is echter ook een van de minst toegankelijke problemen. Onderzoek zoals gedaan door PROCOS is dan ook heel belangrijk, zeker voor op de lange termijn. Er zal echter nog veel onderzoek gedaan moeten worden, voordat er een gedegen oplossing is.

5.5 Wijsheid

Ervaring leidt tot wijsheid. Het is het veiligst om zo bescheiden mogelijke ambities te hebben om succes te verzekeren. Jammer genoeg moedigt software

vaak complexiteit aan door de aanwezige flexibiliteit. Dit moet tegengegaan worden in safety-critical systemen. De safety-critical aspecten van de software behoren ontkoppeld te worden van de minder kritische aspecten indien mogelijk. In dat geval kan er meer inspanning besteed worden aan het verzekeren van de juistheid van de safety-critical onderdelen van het systeem [Bowen, 2000]. Deze ontkoppeling zorgt daarnaast ook voor een transparanter beeld van het systeem en een vermindering van de complexiteit. Het grote geheel wordt nu immers verdeeld in kleine subsystemen, ieder met hun eigen problemen. Er moet echter wel goed in de gaten gehouden worden dat zowel de systemen als de koppelingen tussen de systemen juist werken.

6 Software kwaliteit

Kwaliteit is niet alleen een van de belangrijkste aspecten van software maar ook van een safety-critical systeem. Deze sectie geeft aan waarom het kwaliteitsaspect van software van belang is voor de samenleving. Daarnaast worden een aantal belangrijke aandachtspunten behandeld die naar voren komen bij het gebruik van software in een safety-critical systeem.

Software wordt vaak geplaagd door verschillende kwaliteitsproblemen. Een van de redenen hiervoor is de snelheid waarmee het op de markt gebracht wordt voor commercieel gebruik. Dit leidt vaak tot een product van lage kwaliteit waar verschillende updates en patches voor uitgebracht moeten worden om kwaadaardige software-aanvallen van virussen, wormen of andere vormen van extern hacken te voorkomen. Lage software-kwaliteit is dan ook een ethisch onderwerp voor de samenleving. Het verbeteren van de kwaliteit van software zal gerelateerd moeten worden aan fundamentele ethische richtlijnen. Software ontwikkelaars zullen er op deze manier aan moeten voldoen [Peslak, 2004].

6.1 Software in safety-critical systems

Er is geen reden waarom software niet gebruikt kan worden in bepaalde safety-critical systemen. De software die gebruikt wordt moet echter wel onder bepaalde omstandigheden ontwikkeld zijn. Er is extreme discipline nodig in het ontwerp, de documentatie, het testen en het is nodig dat de software gereviewed wordt. Tevens is het essentieel dat de technieken waaronder de software opereert en de requirements goed duidelijk zijn. Als er niet aan deze condities voldaan is zijn voldoende review en testen niet mogelijk [Parnas, 1990].
Het systeem moet duidelijk gestructureerd worden conform de verborgen informatie zodat het eenvoudiger te begrijpen, te reviewen en te repareren is. De documentatie moet volledig en precies zijn. Er moet gebruik gemaakt zijn van een mathematische notitie in plaats van natuurlijke taal. Iedere fase van het ontwerp moet gereviewed worden door onafhankelijke reviewers met de benodigde gespecialiseerde kennis voor die fase [Parnas, 1990]. Mathematische verificatie technieken moeten gebruikt worden om de review systematisch en strikt te maken. Een onafhankelijk bureau moet statistisch valide rondom testen uitvoeren om schattingen te bieden van de betrouwbaarheid van de kritieke aspecten van het systeem. Uitgebreide kennis en ervaring met het toepassingsgebied is nodig om de inhoud van de test cases te bepalen [Bowen, 2000].

De meeste literatuur over random testing is, voor het grootste gedeelte, niet relevant voor safety-critical situaties. Omdat er geen interesse is in een schatting van de error rates of de oplevering van een betrouwbaarheids-groei-studie is een simpel model voldoende. Hypothese testing maakt het mogelijk om de betrouwbaarheid dat het systeem voldoet aan de requirements te evalueren. De betrouwbaarheid van het systeem op zich wordt nauwelijks in de praktijk uitgevoerd. Het aantal testen dat hiervoor nodig is, is gewoonweg te groot. Betrouwbaarheid moet verkregen worden door gebruik te maken van strikte mathematische technieken in het review proces [Parnas, 1990].
De veiligheid en betrouwbaarheid van het systeem rusten op de aspecten testing, mathematische review en certificatie van personeel en proces. Het aspect certificatie is het probleemaspect. Binnen de software engineering bestaat er geen autoriteit die verantwoordelijk is voor de certificatie van professionele engineers. Engineers zijn dan vaak ook niet voldoende voorbereid om een bijdrage aan een safety-critical systeem te leveren [Parnas, 1990].
Al deze voor safety-critical systemen belangrijke aspecten kunnen beïnvloed worden door een goede ethische code. Deze bepaalt immers de manier waarop engineers een bijdrage leveren aan het systeem. De code zorgt ervoor dat engineers op de hoogte zijn van de gevolgen die een actie kan hebben. Verder is het erg belangrijk dat de betrokken engineers op de hoogte zijn van de denkstijlen die beschreven staan in sectie 5. Wanneer de engineers rekening houden met de daarin beschreven aspecten, dan zullen zij met grotere waarschijnlijkheid de juiste beslissingen nemen met betrekking tot het safety-critical systeem.

7 Ethische richtlijnen bij safety-critical systemen

In deze sectie wordt de informatie verkregen uit de voorgaande sectie aan elkaar gerelateerd en gegroepeerd. Op deze manier ontstaat er een raamwerk wat een overzicht geeft van de belangrijkste aspecten die komen kijken bij de verhoging van de integriteit van mens en systeem.

Het is verstandig om bepaalde richtlijnen te volgen bij het ontwikkelen van een software gebaseerd artefact. Een safety-critical systeem heeft door zijn essentiële verschillen ten opzichte van een standaard systeem dan ook een aangepaste ethische code nodig. De basiscomponenten van een professionele ethische code zijn meestal eenvoudig te begrijpen, de specifieke details vergen echter zorgvuldige aanpassing om aan de wensen van een bepaald beroep te voldoen. De complexiteit van dit proces is waarschijnlijk de reden waarom de codes zo weinig herzien worden. Voorgaande secties hebben verschillende aspecten behandeld die een rol spelen bij het opstellen van een ethische code voor safety-critical systemen. In sectie 5 staan enige door het Hazards Forum[6] opgestelde richtlijnen die specifiek gericht zijn op safety-critical systemen. Deze zouden op zichzelf al gebruikt kunnen worden als ethische code. Om echter een uitgebreider beeld te krijgen worden alle, in voorgaande secties behandelde, aspecten gecombineerd, zodat er een raamwerk gemaakt zal kunnen worden waaraan een ethische code speciaal opgesteld voor een safety-critical system zou moeten voldoen. De door het Hazards Forum opgestelde richtlijnen zullen onder andere in dit raamwerk verwerkt worden. Het raamwerk geeft weer welke eigenschappen van essentieel belang zijn. Als basis voor dit raamwerk wordt de indeling van de ethische code opgesteld door de ACM gebruikt.
De ethische code bestaat uit de vier secties bestaande uit 24 imperatieven[7]. Deze zijn geformuleerd als verklaringen van persoonlijke verantwoordelijkheid. In combinatie met deze imperatieven is er een set van richtlijnen die het individu sturen met betrekking tot de interpretatie en het gebruik van de opgestelde code bij het nemen van beslissingen [Martin, 1998]. Aspecten die extra van belang zijn bij safety-critical systems zullen in de daarvoor relevante sectie genoemd worden.

7.1 Algemene morele imperatieven

De eerste sectie bestaat uit algemene morele imperatieven die een persoon sturen om professioneel gedrag toe te passen. In deze sectie zullen aspecten genoemd worden welke van belang zijn bij het maken en gebruik maken van een ethische code.

[6] The Hazards Forum, *Safety-related systems: Guidance for engineers*, 1995
[7] Zie sectie 3.

Het aantal systemen dat beschouwd wordt als safety-critical blijft toenemen. De lage hardware kosten, de toename van kwaliteit van hardware en andere technologische ontwikkelingen zorgen ervoor dat er veel nieuwe software systemen gebruikt zullen gaan worden in verschillende domeinen. Bij dit alles is het wel belangrijk om op te merken dat software beslissingen vaak genomen worden op basis van economische factoren in plaats van veiligheids aspecten [Bowen, 2000]. Dit is natuurlijk geen goede insteek. De ethische code moet de beoefenaar dan ook in staat stellen om juiste beslissingen te nemen. Verder moet het een bron zijn van professionele socialisatie, het moet beoefenaars helpen om sociaal te worden met betrekking tot hun shared values en doelstellingen van hun beroep. Dit is een algemeen doel van een ethische code wat dus in elke situatie moet gelden. Een situatie waarbij een safety-critical systeem betrokken is vereist het inachtnemen van de volgende zaken.

Het is belangrijk rekening te houden met de huidige beperkingen van software engineering. Engineering van safety-critical systemen is een complexe taak die betrekking heeft op verschillende technische gebieden. Software is een kern component van elke safety-critical systeem. Er zijn echter veel te weinig ontwikkelaars in andere disciplines die begrijpen wat software wel en niet kan. Mensen moeten tijdens hun opleiding al op de hoogte gesteld worden van de aspecten die komen kijken bij safety-critical systemen [Knight, 2002]. Een belangrijk probleempunt is de mate van abstractie die nodig is bij de beschrijving van een systeem. Helaas moeten veel programmeurs, wanneer ze het systeem als een geheel beschouwen, hun natuurlijke wil, om te veel details bij de implementatie te betrekken, afleren. Alleen de relevante aspecten moeten aanwezig zijn bij een bepaald abstractieniveau, dit is vaak niet het geval. Of er zijn teveel details aanwezig, of belangrijke details ontbreken [Bowen, 2000].

Een ander aspect dat problemen op kan leveren is het begrip van het systeem. Het is wenselijk er voor te zorgen dat de scheiding tussen theorie en praktijk geminimaliseerd wordt. Dit is zeker extreem belangrijk op het gebied van safety critical systems. Een goede theoretische en wiskundige onderbouwing is essentieel om een maximaal begrip van het systeem dat ontwikkeld wordt te krijgen. Wanneer dit begrip er niet is kunnen er serieuze problemen eenvoudig voorkomen.

Verdere professionele ontwikkeling is dan ook cruciaal om kwalitatief werk te leveren. Er zijn verschillende cursussen aanwezig om de Master of Science titel te

behalen in safety-critical systems[8]. Een engineer kan op deze manier een certificaat behalen op het gebied van safety-critical systems. Hij toont hiermee aan dat hij competent is voor het betreffende werk. Een dergelijke cursus zou verplicht moeten zijn voor mensen die een bijdrage leveren aan een safety-critical systeem. De inhoud van deze cursus zal goed in de gaten gehouden moeten worden, zodat deze in overeenstemming met met de werkelijkheid blijft. Het curriculum van de cursus zal dus dynamisch ingericht moeten worden, zodat het eenvoudig is om veranderingen te implementeren. Engineers zullen ook aanvullende cursussen moeten volgen, indien er belangrijke veranderingen in het vakgebied zijn.

7.2 Specifieke professionele verantwoordelijkheden

De tweede sectie behandelt meer specifieke professionele verantwoordelijkheden die geïnterpreteerd moeten worden aan de hand van het beroep. In deze sectie zullen aspecten genoemd worden welke van belang zijn bij het maken en gebruik maken van een ethische code.

Bij het ontwikkelen en bouwen van een safety-critical systeem komen verschillende problemen naar voren. Er zullen belangrijke keuzes gemaakt moeten worden om tot een respectabel eindproduct te komen. Aspecten waar de computer engineer rekening mee moet houden zijn onder andere de volgende.
Het falen in de interactie tussen software en systeem engineering veroorzaakt een groot gedeelte van de fouten. Het is van essentieel belang dat veelomvattende benaderingen voor totaal systeem modellering ontwikkeld worden zodat de eigenschappen van volledige systemen geanalyseerd kunnen worden. Een dergelijke benadering moet software op de juiste manier afstemmen en moet betrouwbare modellen van kritieke software karakteristieken bieden [Knight, 2002].
Aspecten die het maken van betrouwbaren modellen moeilijk maken zijn de volgende. Het aantal safety-critical systemen dat met elkaar communiceert in een enkele applicatie zorgt ervoor dat er resources gedeeld moeten worden tussen systemen. Dit elimineert het architecturale element, fysieke separatie, dat vertrouwen moet geven voor een correcte werking. Het falen van een systeem mag immers geen invloed hebben op het correct werken van een ander systeem. Dit gaat verloren wanneer er veel interactie is tussen verschillende systemen, ze

[8] University of York, The, Safety critical systems engineering, system safety engineering: Modular MSc, diploma, certificate, short courses, 2005, The University of York, Heslington, U.K., http://www.cs.york.ac.uk/MSc/SCSE/

worden dan afhankelijk van elkaar. Dit is zo wanneer meerdere functies gesitueerd zijn op een platform zodat de constructie eenvoudiger is. Technieken die hoge zekerheid bieden van non-interventie zijn dan ook nodig [Knight, 2002]. Fouten in de software specificatie spelen een rol in vele vormen van falen. Vele vormen van de specificatie worden niet ondersteund door de huidige technieken. En waar deze technieken al bestaan, daar ontbreekt de integratie om een volledige specificatie analyse te maken. Verificatie door middel van testen is onmogelijk voor systemen die werken op een hoog afhankelijkheids niveau. Formele verificatie en model checking worden dan ook vaak gebruikt bij dergelijke systemen. Deze zijn echter gelimiteerd in hun mogelijkheden. Software verificatie en validatie kunnen extra betrouwbaarheid geven maar metingen van de effectiviteit van dergelijke benaderingen zijn moeilijk te verkrijgen. Dit betekent dat het volledig steunen op dergelijke software een gewaagde keuze is [Knight, 2002].
Gegeven al deze problemen van software zal er bij het bepalen voor het gebruik van software in safety critical systems goed nagedacht moeten worden over de software keus. De keus moet immers gemaakt worden met als doel het verkrijgen van een zo hoog mogelijke zekerheid dat het systeem niet faalt. Het is dan ook aan te raden back-up systemen te gebruiken die erg eenvoudig werken waar de gekenmerkte problemen niet voor kunnen komen. De huidige software is vaak erg complex, waardoor intuïtie[9] zo moeilijk te verkrijgen is in de meeste gevallen. Dit leidt er toe dat er geen validiteit verkregen kan worden of de software voldoet aan de specificatie. Het kernpunt bij dit alles is om simpele software te maken die complexe problemen het hoofd biedt. Dit moet ook wel aangezien de huidige ontwikkelingstijd en moeite om safety-critical systemen te bouwen zo extreem is dat met de huidige technologie het bouwen van toekomstige systemen in veel gevallen niet mogelijk is. Nieuwe software moet oplossingen bieden voor deze tijd en kosten problemen. Er is hier echt een aanzienlijke verbetering nodig. Het is dus belangrijk dat er veel onderzoek plaatsvindt, waardoor technieken zich verder kunnen ontwikkelen, en nieuwe technieken kunnen ontstaan. Bij het in gebruik nemen van nieuwe technieken moet er echter weer voorzichtig gehandeld worden. Het is van belang voor een engineer om conservatief te werken, doch progressief te denken. Hiermee wordt bedoeld dat de engineer niet te snel gebruik moeten maken van nieuwe technieken, maar wel constant bezig moet zijn met het bedenken van nieuwe oplossingen. Nieuwe technieken kunnen pas worden aangeraden en aangenomen voor het gebruik in safety-critical gebieden, wanneer er voldoende ervaring is verkregen en de baten zijn geanalyseerd. Het is dan ook erg

[9] Het aan kunnen geven waarom een programma correct is aan zijn specificatie.

belangrijk dat de voorschrijvende aanbevelingen in standaarden vaak geüpdate worden, zodat de gegevens zo accuraat mogelijk blijven en iedereen op de hoogte is van de status van bepaalde technieken.
Bij dit alles is het ook belangrijk dat de engineer voldoende zelfkennis en zelfkritiek heeft. Iedere engineer heeft zijn beperkingen, het is belangrijk dat invite hun eigen beperkingen herkennen, als ook die van anderen. Engineers zullen dan ook goed op de hoogte moeten zijn van de denkwijzen beschreven in sectie 5. De zaken die hierin beschreven staan zijn een erg belangrijke factor bij een succesvolle afloop van een project.

7.3 Organisationeel leiderschap

De derde sectie presenteert de imperatieven die een rol spelen bij organisationeel leiderschap. De brede verklaringen relateren aan algemene sociale verantwoordelijkheid, kwaliteit van leven, waardigheid en gebruikersrechten.

De ontwikkeling van safety-critical systemen moet als doelstelling hebben het vermijden van het verlies van menselijk leven of serieuze verwondingen door de relevante risico's te beperken tot een acceptabel niveau. Het is de verantwoordelijkheid van het software engineering team en het management van de betrokken organisatie te verzekeren dat geschikte mechanismen aanwezig zijn en dat deze op de juiste manier gebruikt worden om dit doel te bereiken voor de levensduur van het product [Bowen, 2000].
Met het oog hierop is het belangrijk dat een gemeenschap zichzelf toewijdt aan het initialiseren en ondersteunen van bijkomstige activiteiten, zoals ethische comités en veld onderzoeken, het aankaarten van zaken, het bevorderen van discussies en de behandeling van ethisch gedrag als een belangrijk aspect van de activiteiten van de leden. De leden moeten op de hoogte zijn van het belang om volgens de ethische code te handelen. Hierbij is het belangrijk dat er wel gekeken wordt naar de mate van professionele verantwoordelijkheid van de betreffende persoon of groep. Hoe groter de verantwoordelijkheid hoe meer aandacht er besteedt moet worden aan een duidelijke communicatie van de code.
Het is erg belangrijk dat er niet gestopt wordt bij het enkel en alleen opstellen of aanbieden van een ethische code. Er moet ook voldoende ondersteuning zijn, zodat iedereen zijn vragen met betrekking tot de ethische code kwijt kan[10]. Dit aspect geldt voor zowel het leiderschap binnen de organisatie als het leiderschap van de gemeenschap die de code heeft opgesteld. Voor het leiderschap van de organisatie geldt daarnaast het volgende. Het personeel met de hoogste

[10] The Hazards Forum, *Safety-related systems: Guidance for engineers*, 1995

kwalificaties behoort werkzaam te zijn bij de ontwikkeling van safety-critical applicaties. De betrokken engineers moeten capabel zijn om de benodigde kennis gereedschappen op te nemen en een goed begrip te hebben van de benodigde operaties van computergebaseerde systemen. Het dus erg belangrijk dat het Human Resource Management van zeer goede kwaliteit is bij een organisatie waar mensen worden aangenomen dan wel ingehuurd om een bijdrage te leveren aan een safety-critical systeem. Het personeel met de juiste kwalificaties dient zorgvuldig geselecteerd te worden.
De veiligheid en betrouwbaarheid van het systeem rusten op de aspecten testing, mathematische review en certificatie van personeel en proces. Het aspect certificatie is het probleemaspect. Binnen de software engineering bestaat er geen autoriteit die verantwoordelijk is voor de certificatie van professionele engineers. Engineers zijn dan vaak ook niet voldoende voorbereid om een bijdrage aan een safety-critical systeem te leveren [Parnas, 1990]. Dit kan opgelost worden door in eerste instantie een cursus, zoals beschreven in sectie 7.1 verplicht te stellen. Om ook daadwerkelijk een certificatie methode op te stellen voor engineers, zullen er na het volgen van een goedgekeurde cursus, verschillende vervolgcursussen gevolgd moeten worden. De engineer moet immers op de hoogte blijven van de meest recente technieken. Met meest recente technieken worden hier dan wel de technieken bedoeld die ook daadwerkelijk al goedgekeurd zijn.

7.4 Overeenkomstigheid

De vierde sectie behandelt overeenkomstigheid. De verklaring stelt dat overtredingen inconsistent zijn met het lidmaatschap.

De ethische code moet een bron van publieke evaluatie van het beroep zijn en de reputatie hiervan verbeteren. Dit kan enkel wanneer ongewenst gedrag geminimaliseerd wordt. De ethische code moet dan ook afschrikwekkend werken tegenover onethisch gedrag, dit is bij safety-critical systems erg belangrijk, gezien de mogelijke risico's [Frankel, 1989]. Er moet in de code veel aandacht besteedt worden aan dit aspect. De code zal dan ook erg precies moeten zijn, zodat het voor iedereen die er gebruik van maakt, duidelijk is wat wel en wat niet getolereerd wordt. Tevens zullen er bepaalde straffen moeten staan op het niet voldoen aan de code. Deze straffen zullen gericht kunnen worden op de persoonlijke certificatie van de betreffende persoon. Wanneer zijn certificatie ongeldig wordt verklaard zal hij geen bijdrage meer kunnen leveren aan safety-critical projecten.
Het is erg belangrijk dat mensen niet alleen gestraft worden bij het niet navolgen van de code, maar dat zij ook gestimuleerd worden de code wel te volgen

[Martin, 1998]. De gemeenschap die verantwoordelijk is voor de opstelling van de code moet daarom een ondersteunend systeem (moreel, wettelijk, financieel) bieden aan leden die moeilijke ethische beslissingen moeten nemen. Gezien de belangen die op het spel staan, is het van belang dat elke beslissing zo rationeel mogelijk genomen wordt. Hierbij mogen externe elementen de beslisser niet verstoren in de besluitvorming. De gemeenschap moet diegene dan ook zo goed mogelijk beschermen. Tevens moet het een basis bieden voor de berechting van ethische controverses. Het moet duidelijk zijn op basis waarvan besluiten genomen worden. Op deze manier hoeven engineers niet bang te zijn bestraft te worden ten gevolge van een bepaalde keuze. Wanneer zij volgens de ethische code gehandeld hebben dan zal de gemeenschap hen volledig steunen bij hun beslissingen.

8 Conclusie

Het is onethisch om software te ontwikkelen voor safety-critical systemen, zonder gebruik te maken van de best mogelijke mensen en middelen. Alle software engineers en managers die een safety-critical systeem op een professionele manier willen produceren, moeten ervan verzekerd zijn dat ze de juiste training en competenties voor de taak hebben. Ze behoren de mogelijkheid te hebben om zich, zonder angst voor gevolgen, uit te spreken wanneer ze verwachten dat het onmogelijk of gevaarlijk is om een bepaald systeem te ontwerpen. Het is belangrijk dat organisaties, universiteiten, professionele instituten, overheden en al diegenen die belang hebben bij het welzijn van de samenleving, verzekeren dat er voldoende geschikte mechanismen aanwezig zijn om deze doelstelling te waarborgen. Dit betekent onder andere dat er een geschikte ethische code dient te zijn en dat er voldoende ondersteuning moet zijn bij het gebruik hiervan. Deze code draagt bij aan het nemen van deugdelijke beslissingen op basis van de context en het uiteindelijke doel, waardoor integere beslissingen gemaakt kunnen worden.
Omdat het opstellen van een volledige ethische code een erg complex proces is, is er binnen deze scriptie gekozen om uit te gaan van een bestaande ethische code. Aan de hand van de indeling van deze ethische code, is er een raamwerk gecreëerd waarin een aantal, voor safety-critical systemen, erg belangrijke aspecten bijeengebracht zijn. Hiermee worden meteen ook de verschillen tussen safety-critical en standaard systemen duidelijk. De aspecten die genoemd worden verdienen extra aandacht bij safety-critical systemen.
De eerste sectie bestaat uit algemene morele imperatieven die de persoon sturen om professioneel gedrag toe te passen. De tweede sectie behandelt meer specifieke professionele verantwoordelijkheden die geïnterpreteerd moeten worden aan de hand van de IT professie. De derde sectie presenteert de imperatieven die een rol spelen bij organisationeel leiderschap. De brede verklaringen relateren aan algemene sociale verantwoordelijkheid, kwaliteit van leven, waardigheid en gebruikersrechten. En de vierde sectie tenslotte behandelt overeenkomstigheid. Deze verklaring stelt dat overtredingen inconsistent zijn met het lidmaatschap. De verschillende aandachtspunten in deze secties geven een beeld van de aspecten die extra aandacht vergen bij safety-critical systemen. Bij het creëren van een gezonde basis voor een ethische code speciaal gericht op safety-critical systemen zullen deze aspecten dus in acht genomen moeten worden. Daarnaast is het echter ook belangrijk dat de code goed uitgewerkt is. De code moet aan de volgende punten voldoen.

- Het document moet de beoefenaar in staat stellen om de juiste beslissingen te nemen.
- Het moet afschrikwekkend werken tegenover onethisch gedrag.
- Het moet een bron van publieke evaluatie van het beroep zijn en de reputatie van het beroep verbeteren.
- Het moet een bron zijn van professionele socialisatie, het moet beoefenaars helpen om sociaal te worden met betrekking tot hun shared values en doelstellingen van hun beroep .
- Het biedt een ondersteunend systeem (moreel, wettelijk, financieel) aan leden die moeilijke ethische beslissingen moeten nemen.
- Het biedt een basis voor de berechting van ethische controverses.

Wanneer aan al deze punten voldaan is en er met alle zaken uit het raamwerk rekening gehouden is, dan zal er een kwalitatief goede ethisch code zijn, met een goede ondersteuning.
Uit de verschillende bronnen is naar voren gekomen dat deze ondersteuning immers erg belangrijk is bij het gebruik van een ethische code. Afsluitend kan gesteld worden dat de ethische code die gebruikt wordt voor standaard systemen ook gebruikt kan worden bij safety-critical systemen. Er moeten echter wel enige zaken aan toegevoegd worden, voordat de ethische code goed afgestemd is op de eigenschappen van safety-critical systemen. Wanneer dit is gebeurd kan de code de integriteit van mens en systeem versterken.

Literatuurlijst

[Bowen, 2000] Bowen, J., The Ethics of Safety-Critical Systems, *Communications of the ACM*, 2000 (4).

[Burnet, 1892] Burnet, J., 1892, *Early Greek Philosophy*, London and Edinburgh, Adam and Charles Black.

[Davis, 1991] Davis, M., Thinking like an Engineer: The Place of a Code of Ethics in the Practice of a Profession, *Philosophy and Public Affairs*, 1991 (2).

[Frankel, 1989] Frankel, M.S., Professional Codes: Why, How, and with What Impact?, *Journal of Business Ethics*, 1989 (2).

[Gotterbarn, 1997] Gotterbarn, D. e.a., Software Engineering Code of Ethics, *Communications of the ACM*, 1997 (11).

[Hursthouse, 2003] Hursthouse, R, *Virtue Ethics*, Stanford Encyclopedia of Philosophy, 2003.

[Knight, 2002] Knight, J.C., Safety Critical Systems: Challenges and Directions, *International Conference on Software Engineering*, 2002.

[Leveson, 1993] Leveson, N.G., Turner, C.S., An investigation of the Therac-25 accidents, *IEEE Computer*, 1993 (7).

[Martin, 1990] Martin, C.D., Martin, D.H., Professional Codes of Conduct and Computer Ethics Education, *Social Science Computer Review*, 1990 (8).

[Martin, 1998] Martin, D.C., Deconstructing the ACM Code of Ethics and Professional Conduct, *SIGCSE Bulletin,* 1998 (4).

[Marwedel, 2004] Marwedel, P., Secure and Safety-Critical vs. Insecure, Non Safety-Critical Embedded Systems: Do they Require Completely Different Design Approaches?, *Codes + ISSS'04*, 2004.

[Parnas, 1990] Parnas, D.L. e.a., Evaluation of Safety-Critical Software, *Communications of the ACM*, 1990 (6).

[Peslak, 2004] Peslak, A.R., Improving Software Quality: An Ethics Based Approach, *SIGMIS '04,* 2004.

[Rosenberg, 1998] Rosenberg, R.S., Beyond the Code of Ethics: The Responsibility of Professional Societies, *Computers and Society*, 1998.

[Thomas, 1996] Thomas, M., Formal methods and their role in developing safe systems, *High Integrity Systems*, 1996 (5).