

# De Software Engineering Projecten in 1998: Een evaluerend verslag

Lou Somers & Tom Verhoeff  
Faculteit Wiskunde en Informatica  
Technische Universiteit Eindhoven

Juni 1998

## 1 Inleiding

In het studiejaar 1997/98 is het vak Software Engineering (SE) op een nieuwe manier opgezet en uitgevoerd. Deze verandering is gemotiveerd door de invoering van het 5-jarige curriculum voor Informatica in september 1995, de onderwijsvisite bij Informatica in juni 1996 en advies van de OCI.

Watts Humphrey geeft de volgende compacte definitie van de taak van de software engineer:

“to deliver high-quality software at agreed cost and schedule.”  
[Hum97, p.1]

In de studiewijzer wordt het doel van het vak Software Engineering als volgt beschreven:

Kennis en vaardigheden betreffende het projectmatig ontwikkelen van grotere (niet triviale) software-systemen. Fasering van de lifecycle van een software-systeem. Documentatie van het product en het gevolgde ontwikkelproces. Overzicht van methoden voor de ontwikkeling van grote software-systemen. Object-georiënteerde ontwerp- en ontwikkelmethoden.  
Literatuur: [Som95, Pre96]

Meer dan de helft van onze afgestudeerde informatica ingenieurs raakt in hun loopbaan direct betrokken bij software-ontwikkeling. SE is een ingenieursvak bij uitstek. Er is grote vraag naar software engineers, en dat blijft naar verwachting voorlopig nog wel zo.

We vatten het verleden van het vak SE bij Informatica op de TUE kort samen in hoofdstuk 2. Hoe SE dit jaar is opgezet en verlopen, komt aan bod in hoofdstuk 3. In hoofdstuk 4 poneren we wat ideeën voor volgend jaar. Een lijst van gebruikte afkortingen is opgenomen in Appendix A.

## 2 Verleden

Voorheen waren er in het basisdeel van het informatica-curriculum drie vakken met grotere programmeeropdrachten voor groepjes:

- Programmeren 7 (4 sp, zonder college)
- Opdracht Computer-Architectuur en Systeem-Programmatuur (4 sp, zonder college)
- Software Engineering (3 sp, met college)

Deze opdrachten hadden veelal de volgende kenmerken gemeen:

- Studiefase: derde jaar
- Groepsgrootte: ten hoogste vier studenten
- Looptijd: één trimester
- Begeleiding: één staflid, vast per groep studenten
- Klant: i.h.a. geen
- Werkwijze: ruwe fasering en tijdsplanning op papier (bij SE volgens [HvH90]), verder “improviserend”
- Product: partieel geverifieerd prototype

Het grootste bezwaar tegen deze aanpak is dat de opgedane ervaring niet ‘schaalbaar’ is, d.w.z. niet goed toepasbaar is op grotere schaal. Bij deze kleine omvang zijn er weinig communicatieproblemen, zowel onderling als over de tijd (terugvinden wat je vorige maand had bedacht). De noodzaak tot het schriftelijk vastleggen van afspraken rond productontwikkeling en werkwijze komt niet uit de verf. Gegeven de groepsomvang en looptijd is de omvang van het te maken product dusdanig klein dat men (vaak terecht) het idee kreeg dat het beter door één persoon gedaan zou kunnen worden. Verder zou de overhead om serieus wat aan planning en andere management taken te doen relatief groot zijn.

Het kernprobleem van Software Engineering is samen te vatten in één woord: voorspelbaarheid. Dit probleem wordt in de praktijk nog bemoeilijkt doordat met tien middenmoters het werk van twee toppers gedaan moet worden. Een direct hiervan afgeleid probleem is: risicobeheersing. Onder de beste omstandigheden blijkt achteraf dat, zeg, 90% van de gedane moeite overbodig was. Die extra moeite is op te vatten als verzekeringspremie tegen de risico’s. De kunst is die premie goed te besteden. Bovendien worden de taken aan het einde van het traject naar de klant vaak onderschat: het leveren van de *definitieve* versie van programmatuur en handleiding. En juist dat kwam nooit aan bod: bijv. grondig testen werd vanwege de hoeveelheid benodigde tijd meestal achterwege gelaten.

## 3 Heden

### 3.1 Voorbereiding door werkgroep

In de lente en zomer van 1997 heeft een werkgroep zich gebogen over een nieuwe opzet voor Software Engineering. In de werkgroep zaten:

Kees Hemerik	opleidingscoördinator informatica
Johan Lukkien	voorzitter OCI
Ria van Ouwerkerk	externe contacten, EUSTEC
Lou Somers	docent SE
Tom Verhoeff	begeleider SE

Via Pieter Biemond is er inbreng vanuit de studenten geweest.

De werkgroep heeft kennis genomen van twee grootschaligere aanpakken van onderwijs in Software Engineering, te weten de Workshop SE bij de OOTI van de TUE en GiPhouse bij Informatica aan de KUN.

De OOTI Workshop SE (2J249) beslaat circa één trimester, waarin 350 uur per persoon wordt besteed aan één groot project voor een externe klant. Dit jaar bestond het team uit 19 personen met 4 begeleiders van onze vaste staf en liep het project van 13 maart t/m 30 juni. Binnen het team wordt alle productontwikkeling en projectmanagement gedaan. Men schrijft hiervoor een eigen projecthandboek.

Bij Informatica aan de KUN wordt, naast vier college's SE, een Geïntegreerd Practicum (GiP) in vier blokken verzorgd. Het Geïntegreerd Practicum is ondergebracht in *GiPhouse*, een softwarehouse dat volledig draait op studenten (de directeur uitgezonderd). Kenmerken van de vier SE vakken:

**SE1** 4e semester – 3 sp – 5,5 uur/week

**SE2** 5e semester – 2 sp – 4,0 uur/week

**SE3** 6e semester – 2 sp – 5,5 uur/week

**SE4** 7e semester – 2 sp – 5,0 uur/week

Kenmerken van de vier GiP blokken:

**GiP1** 4e semester – 6 sp – 17 uur/week – verplicht

Een gegeven ontwerp [uit GiP2/4] van een groot, gecompliceerd software-systeem ... bestuderen, analyseren en eigen maken. Een dergelijk ontwerp in een samenwerkingsverband realiseren, ... documenteren, testen en invoeren ... onder leiding van een projectmanager en technische manager [uit GiP3].

**GiP2** 5e semester – 5 sp – 14 uur/week – verplicht

In een samenwerkingsverband, de eisen analyseren en definiëren waarvan een in opdracht te bouwen groot software-systeem moet voldoen. [Deze eisen] vertalen in een technisch ontwerp [voor GiP1/3]

... onder leiding van een projectmanager en technische manager [uit GiP4].

**GiP3** 6e semester – 2 sp – 6 uur/week – facultatief

Als manager het voorgegeven ontwerp [uit GiP2/4] ... projectmatig [met GiP1 team] nader doen uitwerken en doen realiseren op kwalitatief voldoende niveau ... binnen redelijk nauw omschreven inhoudelijk kader.

**GiP4** 7e semester – 3 sp – 9 uur/week – facultatief

Als manager voor een realistisch probleem ... projectmatig [met GiP2 team] een ontwerp voor een software-systeem doen vervaardigen op kwalitatief voldoende niveau [t.b.v. GiP1/3].

Bovenstaande informatie is overgenomen uit de on-line vakbeschrijvingen: <http://www.cs.kun.nl/school/opleidingen/vakken/>.

### 3.2 Uitgangspunten voor nieuwe opzet

De werkgroep is destijds gekomen tot de volgende uitgangspunten voor de nieuwe opzet van SE:

1. Opdrachten Programmeren 7, CASP, en SE samenvoegen: één groot SE Project van 10,5 sp (0,5 sp college SE), waarin user-interfaces, databases, CASP en complexere algoritmische aspecten aan bod komen.
2. Grotere teams o.l.v. een OOTI: ten minste acht derdejaars studenten, en één OOTI met ervaring in projectmanagement. Dit levert ten minste  $8 * 10,5 * 40 = 3360$  werkuren per project (circa 2 jaar).
3. Grotere opdrachten van echte klanten voor echte producten: dit bevordert de motivatie en het realiteitsgehalte, en kan eventueel leiden tot een soort *Informatica-werkwinkel*. De klant is geen informaticus en bemoeit zich niet met de uitvoering van het project; de klant kan wel bijgestaan worden door experts die adviseren m.b.t. technische zaken.
4. Projectmatig volgens 'handboek': elk team schrijft eigen projectplannen volgens de ESA SE Standaard t/m TR-fase (zie ook volgende paragraaf) en vult zelf alle projectrollen in. Zo krijgen de studenten beter inzicht in de werkwijze; men is eerder geneigd zich aan zelf-opgelegde regels te houden.
5. Verscheidene experts: stafleden worden als consultants en reviewers ingezet, op basis van nacalculatie (omdat niet op voorhand duidelijk was hoeveel tijd het zou kosten).

6. College SE vooraf in blokvorm: vijf middagen van drie uur met aantekeningen op het web.
7. Plenaire slotbijeenkomst: met presentaties en een feestelijk tintje

Lou Somers is docent van het hoorcollege SE (2M390). Lou Somers en Tom Verhoeff zijn samen verantwoordelijk voor (vormen de “directie” van) de SE Projecten (2M140). Webpagina’s voor deze vakken zijn beschikbaar op:

`<http://wwwis.win.tue.nl/2M390/>`

Het eindresultaat voor beide vakken wordt bepaald op basis van het projectwerk. Er wordt een door GEWIS opgezette mailing list voor deze vakken gebruikt.

### 3.3 De ESA Software Engineering Standaard

Enige kennis van de ESA Software Engineering standaard is nodig om te begrijpen waarom we gekozen hebben voor het gebruik van deze standaard en om de evaluatie te kunnen appreciëren.

De European Space Agency (ESA) is zowel producent als afnemer van complexe software-systemen. Om grip te kunnen houden op (de kwaliteit van) hun eigen en de uitbestede software-projecten heeft de ESA een Software Engineering Standard (PSS-05-0) opgesteld. De eerste versie van deze ESA SE standaard dateert van 1984 en is afgeleid van IEEE standaarden voor SE. De ESA standaard is neutraal geformuleerd (d.w.z. onafhankelijk van modegrillen) en slechts één keer gering gewijzigd sinds de eerste versie.

Na ontdaan te zijn van ESA-specifieke terminologie, is de ESA SE standaard openbaar gemaakt in [MFM<sup>+</sup>94]. Later zijn aanwijzingen bij de deze standaard separaat verschenen in [MFM<sup>+</sup>96]. De kern van de ESA standaard beslaat circa 80 bladzijden. Een citaat uit het voorwoord:

“These Standards provide a concise definition of how to develop quality software. They are brief and comprehensible, and based upon sound and practical principles. They cover the aspects essential for every project; within these limitations they allow as much choice as possible to the project manager.”

[MFM<sup>+</sup>94, p. xi]

Enkele citaten uit de inleiding:

“This document describes the software engineering standards to be applied for all deliverable [ESA] software developed either in house or by industry.

Software is defined in these Standards as the programs, procedures, rules and all associated documentation pertaining to the

operation of a computerized system. These Standards are concerned with all software aspects of a system, including its interfaces with the computer hardware and with other components of the system." [MFM<sup>+</sup>94, p. xiii]

"These standards do not make the use of any particular software engineering method or tool mandatory." [MFM<sup>+</sup>94, p. xv]

Het is belangrijk zich te realiseren dat de ESA standaard slechts een *kader* aangeeft dat nog nader ingevuld *moet* worden voor gebruik in een software-project.

De ESA standaard schrijft voor dat de software life cycle de volgende fasen omvat (tussen haakjes staat een korte karakterisatie):

**UR:** User Requirements Definition (probleemdefinitie: wat wil de klant, informeel maar wel verifieerbaar)

**SR:** Software Requirements Definition (probleemanalyse: wat moet de software doen, specificatie)

**AD:** Architectural Design (probleemoplossing: hoe zit de software in elkaar)

**DD:** Detailed Design and Production (implementatie en testen volgens V-model, handleiding)

**TR:** Transfer (installatie bij klant, Provisional Acceptance Test)

**OM:** Operations and Maintenance (gebruik door klant, aanpassing)

Het is echter niet nodig deze fasen volgens het zogenaamde waterval model te doorlopen (hoewel wij dat wel aanraden; zie hieronder). Ieder van deze fasen dient separaat gedocumenteerd te worden en het resulterende productdocument dient onderworpen te worden aan een zogenaamde externe technische review, d.w.z. met reviewers die niet tot het projectteam behoren en die het document op technische inhoud toetsen (dus niet alleen op vorm).

Verder dienen de volgende plannen opgesteld te worden (tussen haakjes staan enkele kenmerken):

**SPMP:** Software Project Management Plan (werk verdelen en verroosteren, budgettering, risico-analyse, voortgangsrapportage, tijdschrijven)

**SCMP:** Software Configuration Management Plan (hoe worden alle documenten, code, data in toom gehouden)

**SVVP:** Software Verification and Validation Plan (hoe worden reviews en tests opgezet en uitgevoerd)

**SQAP:** Software Quality Assurance Plan (hoe wordt gecontroleerd of de plannen gevolgd worden)

Ieder van deze plannen kent een aantal fase-specifieke delen. Al deze projectdocumenten dienen (per fase) onderworpen te worden aan een externe technische review.

Qua organisatiestructuur wordt het 'unity-of-command' principe aanbevolen: iedereen rapporteert aan precies één persoon [MFM<sup>+</sup>96, p. 310]. De enige uitzondering hierop kan de kwaliteitsmanager zijn die aan de projectleider rapporteert, maar desgevraagd ook rechtstreeks aan de directie.

### 3.3.1 Onze richtlijnen bij de ESA SE standaard

Op de webpagina's is een aantal richtlijnen gegeven voor de SE Projecten. Iedere projectleider rapporteert aan de directie tijdens regelmatig terugkerende voortgangsbesprekingen (ten minste eens per 14 dagen). De directie woont wel externe reviews bij, maar treedt niet op als reviewer. Alle te bespreken documenten (bijv. bij voortgangsbesprekingen en reviews) moeten tijdig op papier aangeleverd worden.

We hebben de voorwaarde gesteld dat al onze SE Projecten tot en met de TR fase (overdracht) volledig worden uitgevoerd. Inkorten van een project (fasen weglaten) is niet toegestaan; desnoods kan men een project afslanken (eisen weglaten) om het alsnog haalbaar te maken. We raden ten sterkste aan om een waterval model te hanteren, gezien de korte doorlooptijd van het project en omdat dit een nettere structuur geeft. In korte tijd incrementeel of evolutionair ontwikkelen leidt al gauw tot hacken. Het ontwikkelen van een prototype wordt wel aangeraden, maar de resulterende code mag niet hergebruikt worden.

## 3.4 De drie projecten

In het studiejaar 1997/98 waren er 28 studenten die het vak SE wilden voltooien. Er moesten dus drie SE projecten gaan draaien. We hebben daartoe vooraf enkele potentiële klanten benaderd en hun mogelijke opdrachten besproken. Op zich was er niets op tegen om meer dan één project te baseren op dezelfde opdracht. Uiteindelijk hadden we echter drie klanten met ieder één geschikte opdracht. De klanten hebben hun opdrachten gepresenteerd op het derde SE college (16 december 1997). Het college werd afgesloten met een open inschrijving op de opdrachten. Vervolgens hebben we zo goed mogelijk drie projectteams gevormd. De projectteams zijn feitelijk pas half januari van start gegaan. De harde einddatum was vastgesteld op vrijdag 12 juni 1998, één week voor aanvang van de tentamens van het lentetrimester.

Met de OOTI coördinator, toentertijd Huub Schols, is overlegd over de inzet van OOTI's, die hiervoor beoordeeld zouden worden t.b.v. hun eigen

opleiding. Het bleek dat er slechts twee OOTI's beschikbaar waren om projectmanagement taken te verrichten bij de SE projecten. Eén van hen zou tot 1 maart beschikbaar zijn, de ander tot 1 juni. We hebben besloten om iedere OOTI bij slechts één project in te zetten. Daardoor moest er dus een groep zonder OOTI gaan draaien. De ChemSoft groep bleek aanvankelijk het meest voortvarend in het zelfstandig uitvoeren van hun management taken, en werd daarom uitgekozen om zonder OOTI verder te gaan.

Alle opdrachten waren "vaag" voor de teams en boden genoeg gelegenheid om een grote groep goed bezig te houden. Volgens de ESA standaard dient de klant het User Requirements Document (URD) te schrijven. Het URD is voornamelijk door de projectgroepen zelf geschreven, uiteraard in nauw overleg met de klant en diens technische experts. Het feitelijke project gaat pas van start als de URD en alle relevante management plannen goedgekeurd zijn. Deze voorbereidende fase kostte bij alle groepen meer doorlooptijd dan verwacht (de totale hoeveelheid bestede tijd in die periode was wel volgens planning). Bovendien waren de plannings al krap m.b.t. testen.

Voor de SE projecten is een overlegruimte (HG 6.05) met een X-terminal aan svstud en afsluitbare archiefkastjes beschikbaar gesteld. Deze ruimte moest in het lentetrimester gedeeld worden met de OOTI's. Voor projectmanagement is door de faculteit MS Project 98 aangeschaft en geïnstalleerd op een PC bij de OOTI's (de OOTI's konden de software gebruiken in ruil voor het beschikbaar stellen van de PC aan de SE Projecten). Vanaf begin april zijn twee kleine werkruimtes (de "onderwijskamers" HG 8.83 en 84) met enkele (oude) "486" PCs continu gereserveerd voor de SE Projecten. Tevens is begin juni twee middagen een zaal met Pentium PCs in het Rekencentrum gereserveerd voor testen.

De studenten moesten de kosten voor afdrukken en kopiëren zelf dragen, maar de faculteit heeft zich garant gesteld als deze kosten aantoonbaar uit de hand zouden lopen. Als reviewers en SE experts zijn ingezet:

Peter Veltkamp  
Marc Voorhoeve  
Jaap van der Woude

Op 27 maart en 3 april heeft ruim twintig van de projectleden een speciale PSP-cursus [Hum97] gevolgd, die extern verzorgd werd op de TUE.

Op 17 juni 1998 is van 15:00 tot 17:00 een slotbijeenkomst georganiseerd. Hiervoor zijn teams, klanten, experts, directie, en andere geïnteresseerden uitgenodigd. De zaal Auditorium 11 was aan de krappe kant. Ieder team heeft een presentatie van een half uur gehouden, inclusief een korte demo. Na afloop was er een borrel.

Hieronder volgen nog wat details over de drie projecten.



### 3.4.1 ChemSoft bij T: Virtuele bouwdoos voor grote molekulen

**Klant:** Marcel van Genderen (TOC: Organische Chemie)

**Groep:** 9 studenten

**Projectmanager:** Steven de Brouwer (student), later Robert Brouwer (student)

**Technisch expert van klant:** Maurice Baars (TOC), Rudolf Mak (INF, alleen in UR fase)

**Platform:** Windows 95

**Gereedschap:** Delphi

**Omschrijving:** Een chemicus kan molekuulmodellen bouwen van balletjes en stokjes, maar voor grote molekulen is dat niet meer interessant. Er bestaat ook software om (virtuele) modellen van molekulen te bouwen, manipuleren, analyseren, weergeven en bewaren. De bestaande software heeft echter enkele beperkingen, zoals het continu afdwingen van fysisch correcte configuraties. Dit bemoeilijkt het opbouwen van grote molekulen. De klant wenst software om virtuele modellen voor grote organische molekulen te bouwen, manipuleren en bewaren (met name in een standaard formaat).

**Opleverdatum:** 15 juni 1998

**Kanttekeningen:** Een groot aantal teamleden heeft nog een studieachterstand. De klant wist aanvankelijk niet zo goed wat de gewenste kernfunctionaliteit van het product zou moeten zijn. Het project heeft geleden onder de wisseling van projectmanager. Bij dit team is eind maart een 'audit' gehouden, waar ook de PP1-coördinator bij was. Het product is wel geaccepteerd door de klant, maar de functionaliteit was ook flink teruggeschoefd.

### 3.4.2 Mercurius bij I: Simulator en navigator voor treintjesbaan

**Klant:** Ria van Ouwerkerk (INF, voorlichting)

**Groep:** 9 studenten

**Projectmanager:** Ronald Koster (OOTI), later Piet de Louw (student)

**Technisch expert van klant:** Eric Verbeek (INF)

**Platform:** Solaris

**Gereedschap:** gnu C++

**Omschrijving:** Informatica beschikt over een computergestuurde treintjesbaan. Hiervoor is reeds software (de blokserver) geschreven die veilig (botsingsvrij) aansturen van de treintjes op "middelhoog" nivo mogelijk maakt (in de trant van "verzoek om trein  $t$  met snelheid  $s$  naar baanvak  $b$  te rijden"). Bij de voorlichtingsactiviteiten wil men scholieren logistieke problemen bieden die ze op kunnen lossen door dienstregelingen te maken. De klant wenst software (de navigator) waarmee scholieren interactief dienstregelingen kunnen samenstellen en uitproberen, en software (de simulator) om fouten (zoals botsingen) te kunnen demonstreren.

**Opleverdatum:** 10 juni 1998

**Kanttekeningen:** De OOTI heeft tot 1 maart het project geleid. De navigator is niet opgeleverd, aangezien de code niet volledig getest kon worden. De simulator is geaccepteerd door de klant. De opgeleverde simulator is al gebruikt bij het afronden van een 'klassieke' SE opdracht voor wiskundestudenten, toen de computer die de treintjesbaan aanstuurt defect bleek te zijn. De software van de wiskundestudenten kon prima gedemonstreerd worden met de gesimuleerde baan.

### 3.4.3 SimPeL bij W: Automatisering van bedrijfssimulatie

**Klant:** Peter Renders (WPA: produktietechnologie en automatisering)

**Groep:** 10 studenten

**Projectmanager:** Bas Bergevoet (OOTI)

**Technisch expert van klant:** Henk van Rooij (WPA)

**Platform:** Windows 95

**Gereedschap:** Delphi

**Omschrijving:** Werktuigbouwkunde-studenten doen bedrijfservaring op in een spel, waarbij een bedrijf met een aantal dochterondernemingen wordt gesimuleerd. Ieder groepje W-studenten beheert één dochter, waarvoor ze orders verwerken (inplannen op machinepark, voorraad regelen, offertes uitbrengen, etc.) De plannings, voorraadmutaties, offertes, etc. worden op papier bij de begeleiders geleverd. De begeleiders houden de huidige toestand (machinebezetting, voorraad, kapitaal) bij, controleren alle voorgestelde plannings, mutaties, etc., en informeren de studenten en directie over het verloop. De klant wenst software om het spel, met name de controlerende en rapporterende functie, te automatiseren.

**Opleverdatum:** 16 juni 1998

**Kanttelingen:** De OOTI heeft tot het einde het project geleid, zij het na 1 juni minder intensief. De klant is zeer tevreden over de werkwijze en het prototype. Het eindproduct is echter niet geaccepteerd door de klant, omdat er nog te veel evidente gebreken waren. Daar tegenover staat dat op de beloofde functionaliteit nauwelijks is ingeleverd. De klant is zeer geïnteresseerd in voltooiing van de software en heeft hiervoor ook een (beperkt) budget.

### 3.5 Beoordelingscriteria

Een aantal problemen waren voorzien met betrekking tot het toekennen van resultaten aan de deelnemers van de SE projecten. Enerzijds zal niet iedereen hetzelfde soort werk gedaan hebben, omdat in het project het werk juist verdeeld moet worden. Anderzijds is moeilijk na te gaan wie nou precies voor wat verantwoordelijk is geweest, omdat een aantal belangrijke beslissingen groepsbeslissingen zijn. Er was daarom afgesproken dat in eerste instantie de projectgroep beoordeeld zou worden met één resultaat (per vak). Slechts in uitzonderlijke gevallen zouden individuen in een groep een afwijkend resultaat krijgen.

Bij de beoordeling van elke projectgroep hebben de volgende aspecten meegespeeld:

**Product** kwaliteit (beloofde versus geleverde: functionaliteit, prestatie, handleiding), kwantiteit

**Project** grip (voorspelbaarheid, risicobeheersing), opzet, efficiëntie, afspraken regelen/nakomen, voortgangsrapportage, oplevering

**Documentatie** technische kwaliteit, netheid, volledigheid, consistentie, tijdigheid

**Mening van klant en experts**

**Omgang met ESA SE Standard**

**Leereffect** In hoeverre is gebleken dat men wat nuttigs geleerd heeft tijdens het project? Zou men het een volgende keer echt beter doen?

### 3.6 Ervaringen

Kort samengevat denken we dat een aantal zaken in de Software Engineering Projecten aanmerkelijk beter aan bod is gekomen dan voorheen. Toch zijn er ook nog zaken die niet lekker zitten.

**Schaalgrootte, voorkennis** Studenten (en ook de meeste stafleden) zijn onervaren met deze schaalgrootte, zowel technisch als organisatorisch; de voorkennis is niet toereikend en dat geldt in feite voor elke fase, maar met name m.b.t. specificeren, ontwerpen en het opzetten en uitvoeren van de tests. Er zit nu een gat tussen de kleine (tweepersoons) projecten van het tweede jaar en de SE Projecten.

**Schrijven eigen 'handboek'** Het is nuttig gebleken om de teams hun eigen handboek te laten schrijven, maar ook erg bewerkelijk. Er waren geen goede voorbeelden voor handen.

**Planningen** De planningen waren op voorhand al krap en zijn niet goed in de hand gehouden. Met name tegen het einde van een project gaat dat wringen, zeker met een harde einddatum.

**Rollen** De klanten en experts hebben niet voldoende inzicht gehad in hun rol.

**Vangnetten** We hebben geen vangnetten kunnen regelen, zoals een specificatie die we van de plank kunnen halen als het team er zelf niet snel genoeg uit komt. Dit blijkt erg veel werk met zich mee te brengen (schaalgrootte).

**Middelen** De beschikbare middelen zijn erg beperkt geweest. De uitrusting van de overlegruimte is ondanks tijdige aanvraag te laat geleverd; de overlegruimte is een aantal weken door het telecollege geblokkeerd; de ondersteunende software is laat geleverd, er was aanvankelijk geen gereserveerde werkruimte, de later beschikbare werkruimte had weinige en oude PCs. Op zich zijn zulke beperkingen niet zo erg (in de praktijk kom je dat ook allemaal tegen), maar het moet niet te ver gaan.

**Kwaliteit en kwantiteit** Alle teams hebben inderdaad hun project tot en met de TR fase voltooid. De omvang en technische kwaliteit van de geleverde producten (hoeveel er geboden wordt voor de geleverde inspanning) valt tegen: 3000 tot ruim 7000 Lines-Of-Code, niet foutvrij; toegegeven, dit is een ruwe en discutabele maat. Met name het testen is niet grondig gebeurd. In de praktijk zou de deadline verschoven moeten worden, maar dat kon bij ons niet.

**Inhoud** Projectmanagement is goed uit de verf gekomen. Wat programmeren betreft zijn zaken als user interfaces, databases en CASP voldoende aan hun trekken gekomen. Complexe algoritmische problemen zijn echter niet aan bod geweest (doel van het vroegere Programmeren 7).

**Inzet, enthousiasme, leereffect** De teams hebben hard en gemotiveerd aan de projecten gewerkt. Men heeft ook aangegeven er veel van geleerd te hebben, al is ons niet altijd duidelijk wat.

**Evaluatierapport** Ieder team heeft een evaluatierapport geschreven.

**Enquête** Er is via het Onderwijs Service Centrum een enquête afgenomen.

### 3.6.1 Opmerkingen van Bas Bergevoet (OOTI)

- De SE Projecten hebben baat gehad bij OOTI als projectmanager.
- De SE Projecten zijn zeer leerzaam voor OOTI's.
- De OOTI dient als projectmanager binnen het team te staan (dagelijks beschikbaar).
- De rol van experts moet niet onderschat worden. Studenten moeten meer kunnen terugvallen op mensen die meer ervaring in software engineering hebben.

### 3.6.2 Opmerkingen van Roel Bouwman (student)

Het SE-project werd door de meeste studenten van te voren gezien als een te groot project, en dan vooral in de zin van het aantal projectleden. Zover als ik het van andere studenten heb gehoord dachten veel mensen van te voren dat het bijna niet te doen zou zijn om met zoveel mensen samen aan een project te werken. Men wist toen ook nog niet dat de project-manager (in twee van de drie teams althans) een OOTI zou zijn die al ervaring had met projecten van dit kaliber.

Die OOTI is dan ook (vooral in de beginfase) enorm nuttig gebleken en ons project lag daardoor snel op de rails. Toen eenmaal de opdracht duidelijk werd en de eerste contacten met de klant gelegd waren was ook iedereen van gedachten veranderd over het voorgaande en nu ziet bijna iedereen het als een zeer nuttige vorm van onderwijs. Het is nu eenmaal heel iets anders en enorm leerzaam om eens een groot project te doen en aan iets te werken waar ook daadwerkelijk een echte klant voor is.

Wel zijn er een aantal kanttekeningen die door de meeste studenten geplaatst worden bij het project. Dit zijn puntjes waar in de toekomst goed over nagedacht moet worden.

Ten eerste de voorzieningen. De PC-zalen die er op dit moment zijn zijn geen goede werkruimte om aan dit soort projecten te werken. Er wordt door de projectleden zo'n 25 tot 35 uur per week aan het project gewerkt, en zeker in de laatste fase van het project is dit bijna alleen maar werk dat achter een PC moet gebeuren. In onze groep werd dan ook op allerlei locaties gewerkt. Dit is echter niet bevordelijk voor de communicatie tussen de

teamleden. In het begin van het project hebben wij dit probleem al aangekaart, maar het bleek op korte termijn niet mogelijk om hier een oplossing voor te vinden. Het lijkt mij dan ook slim om nu al te beginnen met het zoeken naar een oplossing voor volgend jaar. De algemene mening van de studenten die dit project gedaan hebben is dat er per projectgroep een eigen werkruimte zou moeten zijn met voldoende voorzieningen. Wat voldoende is verschilt natuurlijk per project maar ik denk dat dat toch zeker zo'n 3 tot 4 PC's zouden moeten zijn.

Een leuke oplossing hiervoor zouden natuurlijk kamers zijn die in de tweede helft van het tweede en het gehele derde trimester voor software engineering gebruikt worden, en die in de rest van het jaar als kleine PC-zaal beschikbaar zijn.

Verder denk ik dat de ESA standaard op sommige punten te uitgebreid is voor een project van deze omvang. Ook werd de documentatie van die standaard (bij ESA en in het SE-dictaat) door de meeste projectleden als "slecht" omschreven. Hij is vaak te summier en het is moeilijk te snappen wat er nu precies verwacht wordt. Een voorbeelddocument zou hier bijvoorbeeld bij helpen, maar die waren dit jaar nog niet beschikbaar. Volgend jaar zouden hier bijvoorbeeld de documenten van dit jaar voor gebruikt kunnen worden.

In het algemeen denk ik dat het SE-project zoals het er nu ligt een zeer goed vak is, maar dat er voor het volgende jaar nog wel een aantal dingen verbeterd mogen worden, veelal op organisatorisch, maar misschien ook een klein beetje op het inhoudelijke vlak. In het project kwamen ook veel zaken aan bod waar een groot aantal teamleden nog niet veel of geen ervaring mee hadden. Hieronder vallen bijvoorbeeld vergaderen, notuleren, het schrijven van formele documenten, etc. Dit is naar mijn mening een goede zaak en zeker een pre aan projectonderwijs.

## 4 Toekomst

Wij stellen voor om het volgende studiejaar met de Software Engineering Projecten door te gaan op de ingeslagen weg. Er zullen dan een aantal evidente aanpassingen doorgevoerd worden:

**Planning beter afdwingen** We zullen de projectleiders van het begin af strakker de hand laten houden aan hun planning.

**Voorbeelddocumentatie** De documentatie van de afgelopen projecten wordt (deels) beschikbaar gesteld (in lastig te hergebruiken vorm, bijv. PDF).

**Rollen** De rol van klanten en experts wordt ook schriftelijk uitgelegd.

**Vaste experts** Aan ieder project wordt ten minste één stafid als vaste expert gekoppeld.

**Instream** Kwalificaties om mee te mogen doen worden duidelijker vastgelegd en afgedwongen. Bijv. zouden we ook het resultaat van een SE toets (zie hieronder) kunnen gebruiken.

**Meer plenaire bijeenkomsten** Vóór de AD fase wordt ook een plenaire bijeenkomst met presentaties georganiseerd. Dit helpt ook bij het strakker volgen van de planningen en bevordert bovendien de onderlinge communicatie tussen de teams.

**Audits** Bij ieder project wordt ten minste één 'audit' uitgevoerd.

**Vaste werkruimte** Naast een vaste overlegruimte willen we er toch ook voor pleiten dat er per projectteam een vaste werkruimte met ten minste vier goede computers komt.

Er zijn echter ook zaken die nader besproken zullen moeten worden. Hieronder volgen wat voorstellen, waarvan nog niet duidelijk is hoe we ermee moeten omgaan.

**Aansluiting bij andere vakken** Het zaak de aansluiting met andere vakken beter te regelen. Met name is het wat raar dat pas in het 4e jaar een cursus Object-geïntereerd Ontwerpen komt, terwijl dat eigenlijk bij de SE Projecten al hard nodig is.

**Snellere start** Projecten sneller starten, college SE eventueel ermee verweven.

**Projectmanagers** Het is nog niet duidelijk of en, zo ja, hoeveel OOTI's volgend jaar beschikbaar zijn. We kunnen overwegen 4e-jaars studenten in te zetten voor projectmanagement (als stage), maar dan moet daarvoor ook een soort cursus georganiseerd worden.

**Beoordeling verkavelen** Studiepunten in kleinere brokken verdelen en beoordelen. Bijv. college SE afronden met een toets.

**PSP** Wat te doen met PSP?

**Nazorg** Vóóraf iets regelen t.b.v. de OM fase. Bijv. in de vorm van stage of studentassistentenschappen.

**Informatica-werkwinkel** Willen we op termijn een soort Informatica-werkwinkel hebben die kwaliteitsproducten kan leveren aan externe klanten?

We verwachten zo'n 60 à 70 studenten die volgend jaar Software Engineering willen doen. Dit geeft mogelijk een capaciteitsprobleem bij begeleiding en werkruimtes.

## **A Afkortingen**

**AD** Architectural Design (fase in ESA SE standaard)

**CASP** Computer-Architectuur en Systeem-Programmatuur

**DD** Detailed Design and Production (fase in ESA SE standaard)

**ESA** European Space Agency

**GiP** Geïntegreerd Practicum (in curriculum bij Informatica aan KUN)

**IEEE** Institute for Electrical and Electronics Engineers

**INF** Capaciteitsgroep Informatica bij de Faculteit Wiskunde en Informatica

**KUN** Katholieke Universiteit Nijmegen

**OCI** Opleidings-Commissie Informatica

**OM** Operations and Maintenance (fase in ESA SE standaard)

**OOTI** (AIO bij) Ontwerpers-Opleiding Technische Informatica

**PSP** Personal Software Process, zie [Hum97]

**SCMP** Software Configuration Management Plan (uit ESE SE standaard)

**SE** Software Engineering

**sp** Studiepunt (1 sp = 40 belastingsuren)

**SPMP** Software Project Management Plan (uit ESE SE standaard)

**SQAP** Software Quality Assurance Plan (uit ESE SE standaard)

**SR** Software Requirements Definition (fase in ESA SE standaard)

**SVVP** Software Verification and Validation Plan (uit ESE SE standaard)

**TOC** Capaciteitsgroep Organische Chemie bij de Faculteit Scheikundige  
Technologie

**TR** Transfer (fase in ESA SE standaard)

**TUE** Technische Universiteit Eindhoven

**UR** User Requirements Definition (fase in ESA SE standaard)

**URD** User Requirements Document

**WPA** Capaciteitsgroep Produktietechnologie en Automatisering bij de Fa-  
culteit Werktuigbouwkunde



## Referenties

- [Hum97] Watts S. Humphrey. *Introduction to the Personal Software Process*. Addison-Wesley, 1997.
- [HvH90] D. K. Hammer and K. M. van Hee. Fasering en documentatie in software engineering. *Informatie*, 32(2):155–166, 1990.
- [MFM<sup>+</sup>94] C Mazza, J. Fairclough, B. Melton, D de Pablo, A. Scheffer, and R. Stevens. *Software Engineering Standards*. Prentice-Hall, 1994.
- [MFM<sup>+</sup>96] C Mazza, J. Fairclough, B. Melton, D de Pablo, A. Scheffer, R. Stevens, M. Jones, and G. Alvisi. *Software Engineering Guides*. Prentice-Hall, 1996.
- [Pre96] Roger S. Pressman. *Software Engineering: A Practitioner's Approach*. McGraw-Hill, third edition, 1996.
- [Som95] Ian Sommerville. *Software Engineering*. Addison-Wesley, fourth edition, 1995.