

# **Kostenschatten, een methodische aanpak voor Imtech ICT Information Technology**

***‘De kunst van het schatten’***

Bram de Jager  
Universiteit van Amsterdam  
Imtech ICT Information Technology

Afstudeerder:  
Bram de Jager  
0442836

12 augustus 2005

Opleiding:  
1 jarige Master Software Engineering  
Universiteit van Amsterdam,  
Hogeschool van Amsterdam en  
de Vrije Universiteit

Opdrachtgever:  
Imtech ICT Information Technology  
Den Haag  
S. Hadinegoro  
D. Lameir

Afstudeerdocent:  
H. Dekkers

Afstudeerperiode:  
4 april 2005 t/m 22 augustus 2005

## Voorwoord

Voor u ligt het resultaat van de afgelopen drie maanden. Deze maanden hebben voor mij in het teken gestaan van het masterproject van de Master: Software Engineering. Alle bevindingen en ervaringen zijn in dit verslag beschreven. Ik heb met veel plezier gewerkt aan het eindresultaat. Enkel resteert mij het bedanken van een aantal personen die dit mede mogelijk hebben gemaakt.

Mijn dank gaat uit naar Imtech ICT Information Technology, met name Sander Hadinegoro, waar ik mijn masterproject uit heb mogen voeren. Ik heb veel nieuwe praktijk- en theoriekennis opgedaan tijdens het onderzoek. Maar ook op het vlak van het totale proces ben ik met raad en daad bijgestaan.

Andere werknemers binnen Imtech ICT IT wil ik bedanken voor hun deelname aan interviews en enthousiasme dat zij voor het project toonden. Hun bijdrage heeft op verschillende momenten voor een heldere kijk gezorgd.

Tijdens het onderzoek heb ik het genoeg gehad om een interview met Prof. dr. R. J. Kusters af te mogen nemen. Hij heeft mij nieuwe inzichten gegeven die bijdroegen aan een verandering in het begin van het masterproject. Hiervoor mijn dank.

Verder bedank ik mijn begeleidend docent, Hans Dekkers, voor zijn snelle reactie op vragen en het geven van adviezen.

## Samenvatting

Al veertig jaar is kostenschatten een onderwerp van discussie en onderzoek. Iedereen is op zoek naar de 'Silver Bullet' [Brooks 87]. Het zoeken naar een oplossing voor de uitgelopen plannings, projecten die over budget gaan en producten die niet voldoen aan eisen en wensen. [Kusters 99] heeft onderzoek uitgevoerd of de zoektocht naar de 'Silver Bullet' loont en concludeerde dat in de periode van 1988 tot 1998 dat er duidelijk verbeteringen zichtbaar zijn, maar weinig organisaties voldoen aan de voorwaarden van een 'volwassen' kostenbeheersing.

Zo heeft ook Imtech ICT IT, zoals vele bedrijven, geen standaard voor het schatten van softwarebouwkosten. Zij is afhankelijk van de deskundigheid van haar werknemers. Deze manier van schatten van de projectkosten heeft de nadelen dat de schatting subjectief is, de inschatting is niet controleerbaar, er ontbreekt een standaard en is moeilijk overdraagbaar, het uitvoeren van kostenschattingen kan alleen door mensen met schattingservaring worden uitgevoerd.

Binnen de huidige situatie van Imtech ICT IT wordt gekeken naar de invloeden van verschillende activiteiten en factoren op het softwarebouwproces. Welke kostenfactoren spelen een onmisbare rol voor het bepalen van de kosten van een softwarebouwproject. Door de kostenfactoren vast te leggen en het schattingsproces te structureren in een framework de beperkingen van de huidige situatie geëlimineerd.

Het framework beschrijft een standaard aan de hand van een combinatie van kostenmethoden en de situatie van Imtech ICT IT. De schattingsgegevens die het framework vastlegt worden gebruikt om het framework te kalibreren / af te stemmen op de projectspecifieke eigenschappen van Imtech ICT IT. Hierdoor ontstaat een learning-oriented framework dat gebruik maakt van functiepunten en transparant is. Functiepunten maakt een objectieve snelle schatting van de omvang van het systeem mogelijk. En het documenteren en vastleggen van de processtappen zorgt voor de transparantie, wat het framework inzichtelijk en controleerbaar maakt.

## Inhoudsopgave

|  |    |
|--|----|
| Voorwoord .....  | i  |
| Samenvatting .....   | ii |
| 1 Achtergrond en probleemstelling .....                          | 1  |
| 1.1 Achtergrond .....  | 1  |
| 1.2 Theoretische beeldvorming .....                              | 1  |
| 1.3 Probleemstelling / onderzoeksvraag .....                     | 3  |
| 2 Projecten en kosten bij Imtech ICT IT .....                    | 3  |
| 2.1 Projecten bij Imtech .....                                   | 3  |
| 2.2 Winst / verlies projecten .....                              | 5  |
| 2.3 Verkoopproces.....   | 7  |
| 2.4 Strijd realistische schatting en concurrerende offerte ..... | 7  |
| 2.5 Invloeden op een schatting .....                             | 8  |
| 3 Kostenmethoden .....   | 8  |
| 3.1 Classificatie kostenmethoden .....                           | 8  |
| 3.2 De Boehm classificatie .....                                 | 9  |
| 3.3 Welke typering voor Imtech ICT IT.....                       | 11 |
| 3.4 Expertise-based methoden .....                               | 14 |
| 3.5 Learning-oriented methoden .....                             | 16 |
| 3.6 Meeteenheden voor kostenschatten .....                       | 17 |
| 4 Voorstel framework .....                                       | 18 |
| 4.1 Wat is het framework .....                                   | 18 |
| 4.2 Opzet van het framework .....                                | 19 |
| 5 Resultaten .....   | 24 |
| Bronvermeldingen .....   | 25 |
| Verklarende woordenlijst .....                                   | 26 |
| Bijlagen .....   | 26 |
| Functiepuntanalyse.....  | 26 |
| Schattingsformulier .....  | 28 |

## 1 Achtergrond en probleemstelling

De in het plan van aanpak beschreven achtergrond, probleemstelling en onderzoeksvraag worden in dit hoofdstuk beschreven. Wat is de aanleiding voor het onderzoek, wat zegt de theorie hierover en op welke onderzoeksvragen ligt de focus tijdens het onderzoek.

### 1.1 Achtergrond

Imtech ICT IT ontwikkelt applicaties voor het bedrijfsleven. Imtech ICT IT biedt haar diensten aan via drie verkoopkanalen. Het marktlabel Bizzware dat zich richt op de verzekerings- en financiële wereld, Greenware dat zich richt op de agro- en foodsector en Public dat zich richt op de overheid en semi-overheid. De klantenkring is hierdoor divers te noemen op meerdere gebieden. De producten die Imtech ICT IT levert zijn softwarebouw, te verdelen in maatwerk en semi-maatwerk. Maatwerk waarbij een nieuwe applicatie vanaf scratch wordt opgebouwd en semi-maatwerk waarbij standaardmodules aangevuld worden met maatwerk. Naast softwarebouw voert Imtech ICT IT standaard installaties uit.

Voordat een opdracht definitief kan starten dient de offerte door de klant goed te worden gekeurd. Een offerte moet betrouwbaar en overzichtelijk zijn, daarnaast speelt het totaalbedrag een belangrijke rol. Het totaalbedrag is afhankelijk van de urenbepaling. Uren mogen niet te hoog ingeschat worden, want dat resulteert in een afwijzing van de offerte door de klant. Een te laag aantal uren heeft nadelige gevolgen voor Imtech ICT IT zelf, de winstmarge neemt af of er wordt zelfs verlies geleden.

Het bepalen van de kosten voor een project gebeurt op basis van impliciete kennis. Twee technische consultants maken aan de hand van een functioneel ontwerp een kosteninschatting gebaseerd op kennis die zij hebben opgedaan in het verleden. Door beide inschattingen te vergelijken wordt een fixed price bepaald voor een project.

De huidige manier van kostenschatten bij Imtech ICT IT heeft meerdere nadelen:

- De inschatting is subjectief
- De inschatting is niet controleerbaar door de projectmanager
- Voor het maken van inschattingen wordt geen standaard gehanteerd en is hierdoor moeilijk overdraagbaar
- Uitvoeren van kostenschatten kan alleen door ervaren consultants worden uitgevoerd

### 1.2 Theoretische beeldvorming

De hoeveelheid methoden voor het schatten van kosten voor softwarebouwprojecten is overweldigend. Er zijn verschillende methoden die hun plaats als gerenommeerde methode hebben verworven. Op basis daarvan zijn er verschillende afgeleide methoden en methoden door organisaties in huis ontwikkeld. In een groot aantal technische artikelen worden methoden inhoudelijk besproken. [Boehm 00][Briand 98][Briand 99][Leung 01][Liming 97] bespreken manieren om methoden onderling te vergelijken. [Boehm 00] onderscheid zes groepen methoden (typeringen) en geeft een algemeen inzicht in methoden.

Om het aanbod van alle methoden overzichtelijk te maken is een selectie ten behoeve van het verdelen van de methoden in typeringen noodzakelijk. Om een selectie te maken worden criteria opgesteld en op basis daarvan wordt een keuze voor een methode gemaakt kan worden. [Leung 01][Briand 98] en [Briand 99] dragen onder andere automatiseerbaar en interpreteerbaar als criteria aan. [Briand 00] bespreekt criteria op hoog niveau: recentheid, niveau van documentatie, bewijsbaarheid en interpreteerbaarheid. Met deze vier criteria sluit

[Briand 00] een groot aantal methoden uit. De typering die [Boehm 00] en [Briand 00] bespreken in hun artikel vormen samen een classificatie. De classificatie dekt alle bestaande methoden af en is het hoogste niveau voor het bespreken van kostenmethoden. Uit deze bronnen wordt duidelijk dat vanuit de literatuur verschillende criteria op te stellen zijn om een selectie te maken van een aantal overeenkomstige methoden.

*Het bespreken van alle methoden afzonderlijk is onmogelijk vanwege de grote hoeveelheid verschillende methoden. Vanuit een classificatie wordt een onderscheid in verschillende typering gemaakt, die bestaan uit overeenkomstige methoden. De selectiecriteria karakteriseren een typering en selecteren de bijbehorende methoden.*

Het proces 'schatten' kent verschillende activiteiten die stapsgewijs doorlopen moeten worden. Binnen een activiteit zijn verschillende kostenfactoren die invloed uitoefenen op de kosten die voor een activiteit worden gemaakt. [Sommer 04] geeft een algemene introductie van de kosten die gemoeid zijn met een softwarebouwproces. Dit gaat van softwarebouwen tot kosten voor energie en water. [Shepperd 97], [Standish 01], [Levin 03] en anderen bespreken factoren als: accuraatheid, applicatie-expertise, applicatietype, domeinkennis, hergebruik, omvang, productiviteit en ervaringen.

Het proces verdelen in activiteiten wordt besproken door [Leung 01][Levin 03] en [SoftDev 95]. [Levin 03] gaat kort in op de Workbench Breakdown Structure (WBS). [Leung 01] en [SoftDev 95] definiëren beiden zeven stappen die ondernomen moeten worden om tot een schatting te komen. Proces, schatting en / of methode zijn niet los van elkaar te koppelen.

Aan de hand van de theorie is het schattingsproces te verdelen in activiteiten. Binnen de activiteiten zijn de kostenbepalende factoren te benoemen. Met de theorie als basis is een analyse van de huidige situatie bij Imtech ICT IT te maken. Vanuit een classificatie kan een passende typering bij Imtech ICT IT gekozen worden. Naast het bepalen van de typering zijn de kostenbepalende factoren te achterhalen.

*Het schattingsproces is op te delen in verschillende activiteiten. Per activiteit kunnen de kostenfactoren bepaald worden. De activiteiten en kostenfactoren vormen een beeld van het schattingsproces, dat als basis te gebruiken is voor het framework.*

Het omzetten van theorie naar praktijk is de realisatie van het framework. Het opstellen van het framework is enkel mogelijk wanneer een goed beeld van Imtech ICT IT gevormd is. Aan de hand van de opgedane kennis uit de literatuur en interviews wordt de huidige situatie van Imtech ICT IT omschreven.

*Vanuit een classificatie is een passende typering voor Imtech ICT IT te bepalen. Vanuit de typering kan een geschikte methode(n) voor kostenschatten gekozen worden, waarin de kostenfactoren als onderdeel zijn opgenomen.*

De gekozen typering en kostenfactoren vormen de basis voor het nieuwe framework. Vanuit deze basis kan een framework worden opgezet dat aansluit op de softwarebouwprojecten bij Imtech ICT IT.

*Op grond van deze onderzoeken mag men verwachten een framework voor het kostenschatten voor softwarebouwprojecten voor Imtech ICT IT op te kunnen zetten.*

### 1.3 Probleemstelling / onderzoeksvraag

De in paragraaf 1.1 opgesomde problemen zijn de aanleiding voor onderzoek naar een framework ofwel een standaard voor het inschatten van kosten voor een softwarebouwproject. Het framework moet een hulpmiddel zijn voor het berekenen van het benodigd aantal uur dat nodig is voor een softwarebouwproject. Het framework moet voorzien in het objectief uitvoeren en inzichtelijk maken van kostenschattingen. Door gebruik te maken van het framework wordt het mogelijk dat meer mensen kostenschattingen kunnen maken, zonder de kennis die bij de huidige methode nodig is. Daarnaast wordt het voor een projectmanager mogelijk een controle uit te voeren op de inschatting.

De onderzoeksvraag kan als volgt geformuleerd worden:

*Welke factoren spelen een rol bij het schatten van de kosten voor de technische realisatie van softwarebouwprojecten en hoe zijn deze te plaatsen in een passend framework voor Imtech ICT IT?*

Daarbij worden de volgende deelvragen gesteld:

- Wat is de huidige werkwijze, van een technisch consultant, voor het inschatten van de kosten voor softwarebouwprojecten?
- Welke gangbare kostenschattingsmethoden bestaan er voor de technische realisatie van een softwarebouwproject?
- Welke kostenfactoren spelen een rol binnen de gangbare kostenschattingsmethoden?
- Hoe verhouden deze kostenfactoren zich ten opzichte van Imtech ICT IT?
- Is er aan de hand van de kostenfactoren een methode te kiezen of een framework op te zetten voor een technisch consultant?

## 2 Projecten en kosten bij Imtech ICT IT

Dit hoofdstuk geeft de praktijk situatie van Imtech ICT IT weer. Welke projecten worden uitgevoerd en welke invloeden zijn van toepassing hierop.

### 2.1 Projecten bij Imtech

#### Projecttypen

Imtech ICT IT speelt in op de vraag van de zakelijke markt. Per jaar worden gemiddeld vijftig tot zestig projecten uitgevoerd. De projecten verschillen van doorlooptijd, omvang, type, etc. De doorlooptijd van projecten kan drie à vier weken tot maanden of zelfs jaren in beslag nemen. Een indicatie van de omvang van projecten is derhalve moeilijk te geven, want er worden meerdere programmeertalen gebruikt en verschillende typen projecten erkend. Projecten zijn te typeren als maatwerk, semi-maatwerk of een standaard installatie. Bij maatwerk wordt een nieuwe applicatie vanaf scratch opgebouwd en bij semi-maatwerk worden standaardmodules aangevuld met maatwerk. De vergelijkbare typering volgens [Standish 01] zijn hieronder weergegeven bij de softwarebouwprojecten van Imtech ICT IT. De laatste kolom is een indicatie van Standish waar hij een afgeronde procentuele verdeling geeft van de in 2001 in Amerika uitgevoerde softwareprojecten.



| <i>Type Imtech ICT IT</i>                            | <i>%</i> | <i>Type volgens Standish</i>                                     | <i>%</i> |
|--|----------|--|----------|
| Maatwerk<br>(Nieuwe .NET applicaties)                | 40       | Development from scratch using traditional languages and methods | 33       |
|  |          | Development from scratch using an object model                   | 13       |
| Semi-maatwerk<br>(SmartSite, losse .NET componenten) | 40       | Purchased application & modified                                 | 15       |
|  |          | Development some components and purchased others                 | 13       |
|  |          | Purchased application & modified extensively                     | 12       |
| Standaard installatie<br>(Sharepoint, SmartSite)     | 20       | Purchased components & assembled the application                 | 9        |
|  |          | Purchased application & performed no modifications               | 5        |
|  | 100      |  | 100      |

Tabel 1: Typering projecten Imtech ICT IT en Standish [Standish 01]

De tabel toont procentueel grote overeenkomsten tussen Imtech ICT IT en Standish. Imtech ICT IT beslaat vrijwel de gehele softwaremarkt en heeft een grote verscheidenheid aan projecttypen. De standaard installatie projecten vallen buiten de scope van de onderzoeksvraag en zullen niet meegenomen worden in verder onderzoek.

#### **Standaarden**

Een grote afzetmarkt heeft consequenties voor het aantal methoden en technieken dat toegepast wordt. Een grote afzetmarkt betekent veel producten en diensten, die elk hun eigen methoden en technieken hebben. Het ontbreken van een focus op enkele methoden en technieken is mede oorzaak voor het ontbreken van standaarden voor softwareontwikkeling. Er zijn weinig standaarden opgesteld voor het procesverloop, het documenteren van of het schatten van het softwarebouwprojecten. Het gebruik van standaarden heeft als voordeel dat de gegevens consistent vastgelegd worden. Hierdoor is het mogelijk deze gegevens te gebruiken voor vergelijkingen of voor het evalueren van de huidige aanpak. Deze gegevens zijn bij Imtech ICT IT enkel op een hoog, abstract niveau beschikbaar.

#### **Projectteam en rollen**

Binnen de projecten zijn rollen voor accountmanager, projectleider, consultant en ontwikkelaar te verdelen. Bij grotere projecten is het mogelijk om een scheiding te maken in functioneel en technische consultants. Hierbij is de functioneel consultant verantwoordelijk voor het functioneel ontwerp en de technisch consultant voor technisch ontwerp. Het voordeel dat hiermee behaald wordt, is het verhogen van kwaliteit door elkaar aan te vullen op functioneel of technisch gebied.

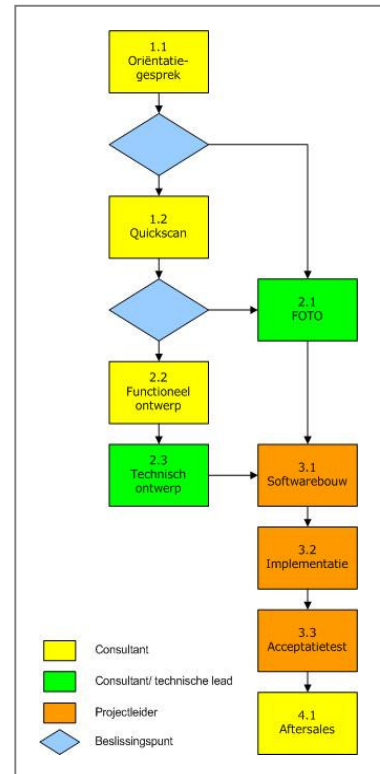
De grootte van een projectteam wordt voornamelijk bepaald door het aantal ontwikkelaars. Een projectteam bestaat uit één accountmanager, één projectleider, één of twee consultants en een aantal ontwikkelaars. Dit kunnen één of meer ontwikkelaar(s) zijn, waarbij het meest voorkomende een maximum zes bedraagt.

## Projectfasen

De softwarebouwprojecten bestaan uit een aantal fasen en elke fase bestaat uit een aantal activiteiten. In figuur 1 zijn de verschillende fasen weergegeven. De kleuren geven de verantwoordelijke persoon aan.

Een project wordt door verschillende factoren beïnvloed. Als de klant een eenduidig eisen- en wensenpakket heeft, kan het mogelijk zijn dat vanuit een oriëntatiegesprek direct een functioneel en technische ontwerp (FOTO) opgesteld kan worden. Bij veel onduidelijkheden kan een quickscan meer duidelijkheid geven, waarna een FOTO opgesteld wordt. Voornamelijk bij grotere projecten wordt gebruik gemaakt van een apart functioneel en technisch ontwerp. Het procesverloop van fase 1 en 2 verschilt hierdoor per project.

Tussentijdse wijzigingen die door klanten worden aangevraagd (request for change, RFC) worden los van het project behandeld. Dit houdt in dat voor een RFC een nieuwe schatting voor kosten gemaakt wordt. Bijbehorende activiteiten worden in een nieuwe planning opgenomen, waardoor een RFC geen invloed heeft op de initiële planning van het softwarebouwproject.



Figuur 1: Projectfasen

## Fixed price of nacalculatie

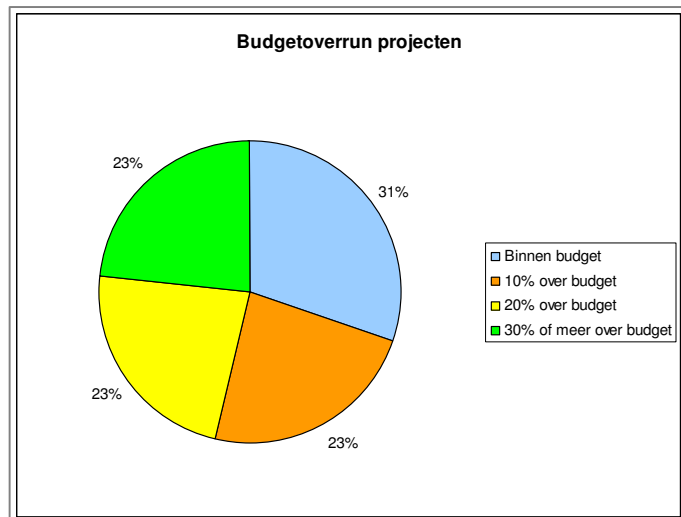
Een schatting wordt gemaakt aan de hand van het technisch ontwerp tijdens fase 2.1 of 2.3. Het technisch ontwerp voorziet in voldoende detail, waardoor een gegronde kostenschatting mogelijk is. Een offerte kan fixed price of op basis van nacalculatie zijn. Een project wordt alleen fixed price uitgevoerd wanneer de kennis van de materie hoog is en er voldoende zekerheid heerst over het behalen van het project binnen de gestelde grenzen. De onzekerheden worden bepaald door aannamen en risico's. Aannamen en risico's worden in een contract opgenomen door middel van clausules in het contract. Hierin zijn de verantwoordelijkheden beschreven van zowel klant als Imtech ICT IT.

## 2.2 Winst / verlies projecten

Uitspraken op grond van empirische gegevens, over het behalen van softwarebouwprojecten, is binnen mijn afstudeerperiode niet mogelijk. Om inzicht te krijgen in winst en verlies zal onderzoek naar schattinggegevens en werkelijk bestede uren uitgevoerd moeten worden. De schattinggegevens zijn op hoog niveau gedefinieerd in de offertes en grotendeels te herleiden. De gedetailleerde schattingen en bestede werkuren vormen het probleem. Gedetailleerde schattingen worden niet centraal, volgens een standaard, gedocumenteerd. De bestede werkuren worden geboekt in een urenregistratie, maar niet alle projecturen worden op het project geboekt. Tijdens het project gemaakte uren worden deels op andere codes dan het uitvoerende project geschreven. Dit zijn studie of andere uren die bij een project ondergebracht zouden moeten worden, maar die door een projectleider buiten het project gehouden worden. Het schrijven van uren gebeurt alleen per project, niet op activiteit. Dus hoeveel tijd aan een activiteit is gependend is niet inzichtelijk. Hierdoor is het vrijwel onmogelijk om een beeld te vormen van de werkelijk bestede uren voor een project. Om toch

inzage te krijgen zijn er aannames gemaakt op basis van een interview met de heer S. Hadinegoro.

De aannames worden gemaakt ervan uitgaand dat de gemaakte projecturen alle uren bevatten die voor het project gemaakt zijn. Dus ook de uren die niet op het project geboekt worden, maar wel binnen het project vallen. 30% van de projecten worden binnen budget afgerond. Deze projecten zijn winstgevend of de kosten en de baten zijn gelijk. Van de 70% die niet binnen het budget afgerond zijn, maakt men een verdeling van eenderde weinig (10%), eenderde veel (20%) en eenderde heel veel (30% en meer) over het budget.



Grafiek 1: Budget-overrun projecten

De schatting speelt hierbij een belangrijke rol, want deze indicatie vormt de basis voor het aantal softwarebouwuuren. Schattingen die Imtech ICT IT maakt zijn reëel naar capaciteit en kwaliteit van ontwikkelaars. Een probleem van een schatting is dat deze niet compleet is. Onderdelen worden vergeten omdat de focus ligt op de schatting voor de softwarebouw en minder op bijvoorbeeld tijd voor testen en bugfix. Naast de realistische schatting staat het proces van het leveren van een marktconforme concurrerende offerte.

Redenen voor het niet afronden van het project binnen budget is niet alleen een verkeerde schatting, maar andere factoren die van invloed zijn, worden weergegeven in tabel 2.

| <i>Project impaired factor</i>        | %    |
|---------------------------------------|------|
| Incomplete Requirements               | 13.1 |
| Lack of User Involvement              | 12.4 |
| Lack of Resources                     | 10.6 |
| Unrealistic Expectations              | 9.9  |
| Lack of Executive Support             | 9.3  |
| Changing Requirements & Specification | 8.7  |
| Lack of Planning                      | 8.1  |
| Didn't Need It Any Longer             | 7.5  |
| Lack of IT Management                 | 6.2  |
| Technology Illiteracy (analfabetisme) | 4.3  |
| Others                                | 9.9  |

Tabel 2: Factoren voor niet slagen softwarebouwprojecten [Standish 95]

### 2.3 Verkoopproces

Zoals eerder aangegeven worden er op jaarbasis vijftig à zestig projecten uitgevoerd, maar niet alle offertes die Imtech ICT IT uitschrijft zullen geaccepteerd worden door klanten. Er worden per jaar tachtig schattingen en offertes gemaakt, dus procentueel wordt ongeveer 30% van de offertes afgekeurd. De redenen voor het afkeuren ligt in eerste instantie bij een te hoog bedrag. Hierbij is de volgende vraag te stellen: biedt Imtech ICT IT marktconforme offertes aan? Het product dat Imtech ICT IT aanbiedt is niet alleen gefocust op een technisch product, maar tevens op het proces dat leidt tot het uiteindelijke product. Imtech ICT IT werkt nauw samen met de klant om gezamenlijk tot een passend eindresultaat te komen. Omdat zij meer aandacht aan het proces schenken, komt dit in de offerte tot uiting in het hogere totaalbedrag.

Hoe wordt dan een opdracht dan verkocht als hij gemiddeld boven de marktconforme waarde ligt? Imtech ICT IT verkoopt het proces met een fixed price voortraject. Het voortraject bestaat uit fase 1 (voorbereiding) en fase 2 (FOTO) van figuur 1. Naast de fixed price voor het voortraject wordt een indicatie, met een grote marge, gegeven van het totaalbedrag. Aan de hand van de indicatie beslist de klant of hij het voortraject wil starten. Na het afronden van het voortraject kan een exacte schatting op basis van het functioneel en technisch ontwerp worden gemaakt en een aanbod aan de klant voor de softwarebouw (fase 3) worden gedaan. Omdat op dit moment het project al gestart is en concurrentie tijdens een eerste indicatie geminimaliseerd of weggenomen is wordt het project voortgezet.

Wanneer de klantverwachting verkeerd wordt ingeschat, omdat deze een technische oplossing wil in plaats van procesmatig tot een gezamenlijk product komen, zal de klant in verhouding een te hoog bedrag worden geoffreerd. Een groot deel van de 30% afwijzingen van offertes hebben dit als oorzaak.

### 2.4 Strijd realistische schatting en concurrerende offerte

Een marktconforme concurrerende offerte en een realistische schatting zijn elkaars tegenovergestelde. Het is een constante strijd tussen het zo laag mogelijk aanbieden van een oplossing en de realisatie ervan. De accountmanager is verantwoordelijk voor de verkoop en richt zich op een concurrerende offerte. De projectleider en het projectteam voeren het project uit. Het gezamenlijke doel is een passende oplossing te vinden zonder verlies te lijden. Het doel wordt ondermijnt door de strijd, want een schatting die realistisch is, maar hoog, levert geen concurrerende offerte op.

Imtech ICT IT is erg afhankelijk tijdens het gehele proces van zijn klant. Het is een zeer dynamische omgeving waarin projectmanagement een belangrijke rol speelt. Fixed price projecten waar nog te veel onduidelijkheden zijn lopen snel uit door de wijzigingen vanuit de klant.

De focus van mijn onderzoek ligt op de schatting die na fase 2 (figuur 1) wordt gemaakt en heeft als input een technisch ontwerp. De schatting moet een realistische schatting zijn, die inzicht geeft in de werkelijk besteedbare uren aan softwarebouw. De invloeden van het verkoopproces en marktconforme offertes zijn voor dit onderzoek niet van toepassing, maar zijn wel van belang voor het uiteindelijk afzetten van de geschatte uren tegen de bestede uren.

## 2.5 Invloeden op een schatting

Het maken van de schatting wordt uitgevoerd door een technische lead. Dit is een ontwikkelaar die vaardig is om een schatting te maken. Hij heeft ruime ervaring opgedaan met ontwikkelen (minimaal vijf à zes jaar) en is bekend met het product of de techniek. De aanpak voor het maken van een schatting is persoonsgebonden. Gemeenschappelijke aanpak is het opdelen van de requirements in hanteerbare activiteiten, zoals ontwikkeling schermen, objecten, functies en tabellen. Aan een activiteit wordt een aantal manuren toegekend dat benodigd wordt geacht voor de ontwikkeling van de activiteit. Voor de schatting van het aantal manuren voor een activiteit wordt impliciet rekening gehouden met de factoren: omvang, complexiteit, datalayer generatie, testen, documentatie en implementatie. Deze factoren hebben allen invloed op de hoeveelheid manuren voor ontwikkeling.

## 3 Kostenmethoden

Het vorige hoofdstuk heeft een beeld van de praktijk situatie weergegeven. Dit hoofdstuk diept de tot nu toe besproken theorieën verder uit en snijdt nieuwe theorieën aan. Hoofdstukken 3.1 tot en met 3.5 beschrijven kostenmethoden vanuit hoog niveau naar laag niveau. De grootheden voor de belangrijkste kostenfactoren komen in hoofdstuk 3.6 aan bod.

### 3.1 Classificatie kostenmethoden

Ten tijde van de start van het onderzoek werd snel duidelijk dat er een overvloed aan methoden is voor het bepalen van de kosten voor softwarebouwprojecten. De belangrijkste conclusie in [Boehm 00] is dat geen enkele methode volstaat in alle situaties. Om een juiste keuze voor een te gebruiken methode te maken, is onderzoek naar en vergelijking van verschillende methoden noodzakelijk. Om overzicht te creëren in de grote hoeveelheid methoden vormen overeenkomstige eigenschappen van methoden een basis voor groepering. Welke methoden tonen dezelfde werkwijze of achtergrond en ideeën? Zijn methoden te groeperen?

Er zijn verschillende onderzoeken gedaan naar de classificatie van methoden. Het voornaamste probleem wat hierbij speelt is de subjectieve manier van het classificeren van de methoden. [Leung 01] verdeelt de methoden in twee typering: algoritmische en niet-algoritmische modellen. Algoritmische modellen lopen uiteen van eenvoudige tot complexe wiskundige berekeningen. Een belangrijke eigenschap van de algoritmische typering is het kalibreren / configureren van de methode aan de organisatie om de accuraatheid te kunnen waarborgen. De algoritmische typering is niet *off-the-shelf* te gebruiken en het kalibreren van de methode aan de organisatie garandeert nog geen hoge accuraatheid. [Briand 00] past een hiërarchische typering toe met twee hoofdcategorieën, model based en non-model based, met verdere verfijning in subcategorieën.

In [Boehm 00] worden zes verschillende typeringen onderscheiden en wordt inzicht in methoden algemeen verworven. De zes typering die hij onderscheidt zijn: *model-based models*, *expertise-based techniques*, *learning-oriented techniques*, *dynamics-based models*, *regression-based models* en *composite techniques*.

Omdat er geen algemene overeenstemming of standaard is voor de classificatie van kostenmethoden en tevens een classificatie als leidraad zal dienen voor het onderzoek zal een keuze voor een classificatie gemaakt moeten worden. Het doel van de classificatie is om alle methoden op te delen in een typering, zodat een top-down (classificatie → typering →

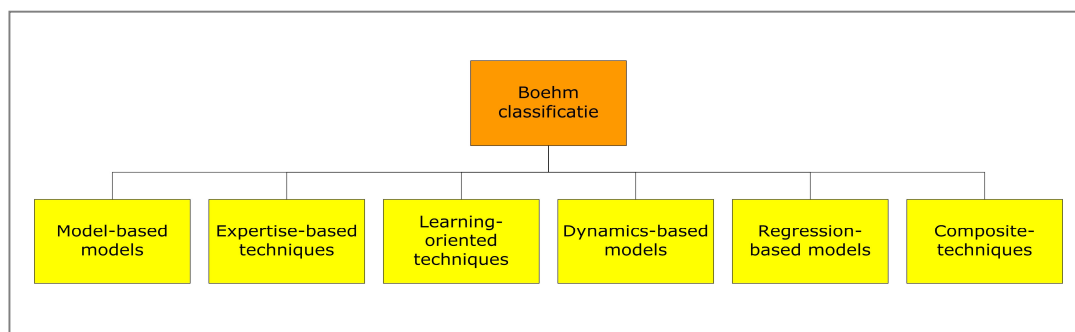
methode) benadering gemaakt kan worden. De classificatie geeft een overzicht voor het maken van een keuze voor een methode die toegepast kan worden binnen Imtech ICT IT.

[Leung 01] maakt enkel verschil tussen twee typeringen, terwijl [Briand 00] en [Boehm 00] de methoden verdelen over vijf of zes typeringen. Beide beschrijven de typeringen uitgebreid en het enige verschil is de hiërarchische opbouw van [Briand 00] tegenover de platte opbouw van [Boehm 00].

Omdat er geen standaard is voor het classificeren van typeringen en het een subjectieve selectie is, zal de keus voor een classificatie op hanteerbaarheid en compleetheid worden gebaseerd. Er is gekozen om gebruik te maken van de classificatie volgens [Boehm 00], omdat Boehm zich richt op de classificatie van kostenmethoden specifiek en dekt hiermee alle kostenschattingsmethoden. De classificatie die Briand heeft opgezet is meer gericht op het ondersteunen van zijn standpunt. De classificatie van Boehm is hanteerbaar, omdat er een duidelijke beschrijving van elke typering wordt gegeven en deze wordt toegelicht met één of meerdere onderliggende methoden.

### 3.2 De Boehm classificatie

De zes verschillende typeringen die [Boehm 00] onderkent worden weergegeven in figuur 2. Het zijn vijf typeringen met hun eigenschappen en één overlappende typering die gebruik maakt van een combinatie van twee of meer typeringen. Wat de verschillen zijn tussen de groepen en welke methoden hierbij horen wordt hieronder besproken.



Figuur 2: Typering methoden [Boehm 00]

#### Model-based models

Afgelopen jaren is veel onderzoek gedaan binnen dit vakgebied en dit resulteerde in vele methoden. Een groot aantal is afkomstig uit onderzoek en is niet te vergelijken vanwege hun structuur. De overeenkomsten waar deze methoden wel op te vergelijken zijn, zijn de in- en uitvoer parameters en de activiteiten van het ontwikkelproces die meegenomen worden in de methode. De zeven populairste methoden hierbinnen zijn: SLIM, Checkpoint, Price-S, Estimacs, Seer-SEM, Select Estimator, Cocomo II.

#### Expertise-based techniques

De op expertise (deskundigheid) gebaseerde technieken zijn nuttig bij het ontbreken aan gekwantificeerde, empirische gegevens. De methoden baseren hun schattingen op de kennis van de experts. Zij zijn nauw betrokken of hebben deelgenomen aan de eerdere overeenkomstige projecten. Deze projecten dienen als basis voor de schatting. Nadeel aan de methode is dat de schatting subjectief is. Controle door middel van nacalculatie op de schatting is onmogelijk en een foute schatting wordt gaandeweg het project pas zichtbaar. Jarenlange ervaringen gaat niet evenredig aan de accuraatheid van de schatting. De Delphi

techniek probeert de subjectieve schattingen te minimaliseren door meerdere experts in het schattingproces te betrekken. De Work Breakdown Structure (WBS) verdeelt het uit te voeren werk op in taken. Zodat per taak een schatting gemaakt kan worden, waarna via een bottom-up het totale budget kan worden berekend.

#### **Learning-oriented techniques**

Technieken die gebruik maken van gegevens uit het verleden komen voor in oude en nieuwe methoden. Om te leren van case studies worden in detail uitgewerkte projecten gebruikt voor het extrapoleren van heuristieken. Vanuit de heuristieken worden verbanden tussen oorzaak en gevolg gelegd. Een stap verder zijn de neural networks die verbeteringen op gebied van het schattingproces proberen te automatiseren aan de hand van de gegevens uit het verleden.

#### **Dynamics-based models**

Softwarebouwprojecten zijn van dynamische aard. Zij zijn aan veranderingen onderhevig en veranderen niet alleen voor de start, maar ook tijdens de uitvoering van het project. Kostenfactoren van deze methoden zijn dynamisch in plaats van statisch en dit vormt het belangrijkste verschil met andere typering. Een belangrijke kostenfactor die wordt beïnvloed is de productiviteit. Hierbij is te denken aan wijzigingen in deadlines, scholing, personeelsbezetting, etc. Veel technieken binnen deze typering zijn gebaseerd op de bevindingen van Jay Forrester, bijna 40 jaar geleden. De methode System Dynamics Approach is goed in te zetten bij het maken van een planning, maar moeilijk om te kalibreren.

#### **Regression-based models**

Regressie technieken worden veelal gebruikt voor het opzetten van een methode vaak in combinatie met model-based techniques. Er wordt onderscheid gemaakt tussen standaard en robuuste regressie. Standaard regressie komt overeen met lineaire regressie en heeft als basis Ordinary Least Squares (OLS). Het is statistisch verband leggen tussen gegevens, zodat een voorspelling gemaakt kan worden. Volgens [Boehm 00] is de methode populair vanwege het eenvoudige gebruik, maar [Briand 00] spreekt dit tegen en geeft aan dat het voor engineers en managers moeilijk te gebruiken is, voor het toepassen is training en ervaring vereist.

Robuuste regressie gaat een stap verder en is een verbetering op het standaard OLS. De verbetering komt voort uit de algemene problemen binnen softwareprojecten: Terminologie zoals definities, verschillende ontwikkelmethoden en het verschil in kwantiteit en kwaliteit met betrekking tot data. De methoden richten zich op het verminderen van de absolute of relatieve errors. Binnen robuuste regressie vallen verschillende technieken onder andere Least Median Squares (LMS) en Least Squares of Balanced Relative Errors (LBRS) [Briand 00].

#### **Composite techniques**

Aan alle bovenstaande typering zijn voor- en nadelen toe te kennen. Om hier optimaal gebruik te maken kunnen verschillende typering gecombineerd worden. De Bayesian analyse is gebruikt voor het ontwikkelen van Cocomo II. Het is een combinatie van gebruikmaken van gegevens uit verleden in combinatie met expertise (Expertise-based). De afwijking van de schatting van beide methoden bepaalt de leidende methode voor de schatting, waardoor de accuraatheid wordt verhoogd.

### 3.3 Welke typering voor Imtech ICT IT

Wanneer de situatie bij Imtech ICT IT wordt vergeleken met de verschillende typeringen in 3.1 dan zullen een aantal typeringen wegvallen en een aantal overeenkomsten vertonen. De eerste selectie elimineert typeringen die afhankelijk zijn van gegevens uit eerdere projecten. Gegevens van projecten uit het verleden zijn, zoals in hoofdstuk 2 beschreven wordt, niet tot slecht beschikbaar. Een typering die gebaseerd is op deze gegevens is moeilijk te kalibreren bij Imtech ICT IT. Binnen de afstudeerperiode is het niet haalbaar om een dataset (gegevens van projecten) op te bouwen. De dataset dient als basis voor de methoden binnen de typering om hieruit informatie te distilleren voor het maken van een schatting. De typeringen *learning-oriented techniques*, *regression-based models* en *composite techniques* hebben alle drie datasets als basis voor hun schatting.

*Dynamics-based models* is een typering die afwijkt van andere typeringen. Deze typering is gericht op wijzigingen die optreden tijdens het project. Schattingen die tijdens het project veranderen en opnieuw geschat moeten worden. De scope beperkt zich op het schatten van de kosten voor de technische realisatie van het softwarebouwproject. De schatting wordt gemaakt tijdens het technische ontwerp en beslaat de technische realisatie. Omdat *Dynamics-based* zich richt op wijzigingen tijdens het project valt deze typering buiten de scope.

De twee verschillende typeringen die overblijven zijn *model-based models* en *expertise-based techniques*. *Model-based* typering waaronder verschillende uiteenlopende methoden vallen. De methoden zijn moeilijk te vergelijken qua structuur maar op in- en uitvoer parameters en de activiteiten van het ontwikkelproces die meegenomen worden in de methode. *Expertised-based* typering maakt gebruik van de kennis van schatters die zij in het verleden opgedaan hebben. Echter jarenlange ervaring staat niet garant voor accuraatheid van de schatting.

In [Jorgensen 04] worden de twee typeringen tegenover elkaar gezet. Jørgensen concludeert dat geen van beide typeringen voorkeur heeft, maar dat de keuze afhankelijk is van afstemming van de methode op organisatie. Wanneer een methode volledig op de organisatie is afgestemd zal de accuraatheid hoger zijn bij *model-based* dan *expertise-based*. Een situatie die meer voordeel voor *expertise-based* oplevert, treed op wanneer specifieke domeinkennis nodig is. De eindconclusie van Jørgensen is:

*"The review concludes that expert estimation is the dominant strategy when estimating the effort of software development projects, and that there is no substantial evidence supporting the superiority of model estimates over expert estimates."* [Jorgensen 04]

Een ander selectiecriteria is acceptatie. In een interview benadrukte prof. drs. R. Kusters dat acceptatie een belangrijk gegeven is voor het nieuwe framework. In [Cooper 99] wordt acceptatie besproken aan de hand van de term cognitieve wrijving. Dit is de weerstand van gebruikers voor een nieuw systeem, software of framework. Wanneer het framework op meerdere manieren geïnterpreteerd kan worden en er geen eenduidige opzet is, treedt cognitieve wrijving op. Gebruikers zullen zich verzetten en het framework (deels of geheel) zal alleen gebruikt worden wanneer er persoonlijk voordeel uit behaald kan worden door de gebruiker.



### Aansluiting typering op Imtech ICT IT

De huidige manier van werken van Imtech ICT IT heeft de meeste overeenkomsten met de typering expertise-based models. Imtech ICT IT baseert haar schattingen op de ervaring van technische leads en laat de schatting uitvoeren door twee technische leads. Voor het beperken van de cognitieve wrijving heeft het gebruik van een expertise-based methode een voordeel. Imtech ICT IT is al bekend met de methode en heeft het vertrouwen van de medewerkers voor een expertise-based methode al gewonnen. Uit de probleemstelling (1.3) wordt duidelijk dat de huidige manier van werken niet voldoet. In tabel 3 worden factoren van de expertise-based typering weergegeven die inzicht geven in de sterke en zwakke punten van de typering.

| <i>Evaluation criteria</i>            | <i>Expertise-based models</i> |
|---------------------------------------|-------------------------------|
| Assumptions<br>(many --, few ++)      | ++ (few)                      |
| Repeatability<br>(low --, high ++)    | -- (low)                      |
| Complexity<br>(high --, low ++)       | + (low/ high)                 |
| Transparency<br>(low --, high ++)     | -- (low)                      |
| Generalizability<br>(low --, high ++) | ++ (high)                     |

Tabel 3: Evaluatie van expertise-based typering [Briand 00]

Binnen expertise-based typering wordt geen gebruik gemaakt van datasets. In plaats daarvan worden schattingen op basis van deskundigheid ofwel expertise uitgevoerd. Aannamen (assumptions) op basis van gegevens van voorgaande projecten worden niet gemaakt, hierdoor wordt hier de waarde ++ aangekoppeld. Dit wil niet zeggen dat er geen aannamen worden gemaakt. Voor verschillende situaties worden aannamen gemaakt, maar dit is niet opgenomen in tabel 3. Dus geeft de waarde voor aannamen een vertekend beeld.

Van schatters wordt verwacht dat zij onbevooroordeeld en accurate schattingen kunnen maken. Dit is onmogelijk, want de mens is altijd bevooroordeeld. Maar door het toepassen van de juiste elicatie technieken kan dit geminimaliseerd worden [Meyer 91]. Elicatie technieken zijn niet alleen van belang voor aannamen, maar in combinatie met gestandaardiseerde documentatie van een schatting wordt het maken van schattingen beter herhaalbaar gemaakt (repeatability). Het herhaalbaar maken van de schatting tot detail niveau is niet mogelijk door het ontbreken van algoritmen, zoals die wel bij de model-based typering aanwezig zijn. Het formeel vastleggen van de stappen voor het bereiken van een schatting zorgt voor het herhaalbaar maken en standaardiseren van het schattingproces. De expertise-based methode Wideband Delphi voorziet in voorgeschreven regels voor het begeleiden van het schattingsproces door middel van het een aantal voorgedefinieerde stappen.

De complexiteit (complexity) van de expertise-based typering is eenvoudig. Het toepassen van de methoden voor een schatting vereist geen specifieke kennis van definities of invoervariabelen voor gebruik van de methode, zoals bij meeste model-based methoden. De complexiteit van expertise-based typering zit in het aantal schatters, de verschillende en aantal overleggen of het aantal voorgeschreven stappen van het schattingsproces.

De transparantie (transparency) van een schattingsmethode is voornamelijk afhankelijk van het gebruik van uniforme definities. Uniforme definities zorgen voor een gedeeld jargon en

maakt schattingen voor iedereen begrijpbaar. De transparantie van de expertise-based typering is laag. Dit hangt samen met het subjectieve karakter en de individuele schatting. Iedere schatter gebruikt zijn eigen jargon, waardoor schattingen moeilijk bespreekbaar zijn en niet transparant. Ook hier geldt het toepassen van juiste elicitatie technieken verhoogt de transparantie.

De laatste factor die wordt geëvalueerd is generaliseerbaar (generalizability). Als een typering toepasbaar is op verschillende omstandigheden, applicatiedomeinen of ontwikkelmethoden dan heeft deze een hoge graad van generaliseerbaarheid. De expertise-based typering wordt veel toegepast in de praktijk. Een onderzoek binnen Nederlandse bedrijven wijst uit dat 37,4% van de organisaties in 1998 gebruik maakt van expertise-based methoden [Kusters 99]. In tien jaar tijd is dit 11,9% gestegen (25,5% in 1988). De typering is in te zetten in verschillende situaties en algemeen bruikbaar.

Het inzetten van de expertise-based typering bij Imtech ICT IT heeft als voornaamste voordelen het invoeren van standaarden voor het schatten. Dit resulteert in een gedeeld jargon en eenduidige set afspraken voor het schattingsproces. De theoretische beeldvorming, van de verschillende methoden die hiervoor ingezet kunnen worden, wordt in hoofdstuk 3.4 uitgediept.

De doelen die gesteld zijn in hoofdstuk achtergrond en probleemstelling worden deels behartigd door de expertise-based typering. De expertise-based methoden voorzien voornamelijk in het structureren van het schattingsproces. De standaarden maken een schatting minder gebiast, waarmee objectiviteit wordt verhoogd. Transparantie is laag, maar door gebruik van elicitatie technieken wordt het proces inzichtelijker en controleerbaarheid vergroot. Maar ondanks deze verbeteringen blijft het schattingsproces zeer afhankelijk van de deskundigheid van de schatter. In de keuze voor een nieuwe methode is Imtech ICT IT op zoek naar een methode die minder afhankelijk is van ervaring.

| *Met minder ervaring een objectieve schatting maken.*

Om dit te bereiken zal een stap verder gezet moeten worden dan alleen het structureren van het schattingsproces. De expertise-based methode moet mogelijk gecombineerd worden met een methode uit een andere typering of worden aangevuld met Imtech ICT IT specifieke factoren. Wanneer wordt gekeken naar de typering volgens de Boehm classificatie komt de learning-oriented typering in aanmerking als aanvulling op de expertise-based typering. De learning-oriented typering kan enkel gebruikt worden, wanneer een organisatie beschikking heeft van een dataset met gegevens uit het verleden. Hoofdstuk 2 geeft aan dat gegevens uit het verleden moeilijk te verzamelen zijn voor het kalibreren van een methode. Echter is het wel mogelijk een dataset op te bouwen. Met de gegevens in de dataset als invoer, kan een iteratief proces gestart worden dat een methode kan kalibreren. Door te leren van de gegevens in de dataset, kunnen aanpassingen aan de methode gemaakt worden voor verbeteringen. Dit zijn de karakteristieken van een learning-oriented methode.

### **Een expertise-based typering aangevuld met learning-oriented typering**

Vanuit de gedachte voor het structureren van het schattingsproces en de huidige manier van werken van Imtech ICT IT wordt een expertise-based typering gekozen. De acceptatie zal het grootste zijn, wanneer een methode wordt gekozen die aansluit bij de huidige manier van werken en schatten. De expertise-based typering voorziet in deze behoefte en vult dit aan met uniforme afspraken, standaarden en een gedeeld jargon. De expertise-based typering wordt aangevuld met een learning-oriented typering voor het afstemmen van het framework aan de organisatie. Met ervaringsgegevens uit toekomstige projecten kan het framework worden gekalibreerd. De combinatie van deze twee typeringen wordt in verschillende artikelen aangehaald als complementaire methoden.

### **3.4 Expertise-based methoden**

Binnen de typering die in 3.3 is gekozen, expertise-based typering, zijn een aantal verschillende methoden ondergebracht. In dit hoofdstuk zullen de drie meest voorkomende expertise-based methoden besproken worden.

#### **Wideband Delphi**

De Wideband Delphi is een groepsconsensus methode die inconsistentie van de schattingen weet te minimaliseren. [Liming 97] pretendeert dat deze methode de beste is, maar geeft aan dat de sterke punten en de zwakheden complementair zijn aan de sterke punten en de zwakheden van algoritmische methoden. Om voldoende communicatie bandbreedte voor de deskundigen te creëren voor het kalibreren van hun schattingen, met die van andere deskundigen, maakte [Boehm 81] in 1981 een voorstel voor het wijzigen van de Delphi methode. Dit resulteerde in Wideband Delphi. De Wideband methode van Delphi erkent dat het oordeel van deskundigen verrassend nauwkeurig kan zijn. Maar de individuen lijden vaak aan onvoorspelbare biases en groepen kunnen "volg de leider" gedrag tentoonstellen. Wideband Delphi pakt beide problemen aan.

De stappen die ondernomen moeten worden voor het gebruik van de methode zijn:

1. De coördinator geeft elke deskundige een specificatie en schattingsformulier
2. De coördinator roept een groepsvergadering uit, in wie de deskundigen schattingskwesties bespreken met de coördinator en elkaar
3. De deskundigen vullen anoniem formulieren in
4. De coördinator bereidt en verspreidt een samenvatting van de schatting op een herhalingsformulier
5. De coördinator roept een groepsvergadering uit, die speciaal concentreert op het bespreekbaar maken van de sterke verschillen in de schattingen
6. De deskundigen vullen opnieuw formulieren in (anoniem) en stappen 4 tot en met 6 worden meerdere malen herhaald tot de schatting op wenselijk niveau is

De methode is succesvol door de combinatie van de vrije discussie voordelen tijdens groepsvergaderingen en de anonieme schattingen van de standaard Delphi methode.

De voordelen van deze methode zijn:

- Het bespreekbaar maken van verschillen tussen opgedane projectervaring en beeldvorming van het voorgestelde project
- Het door de deskundigen bespreekbaar maken van de impact op het project door nieuwe technologieën, architectuur, toepassingen en talen. Hierbij kan ook gedacht worden aan uitzonderlijke personeelskenmerken en interactie, enz

De nadelen zijn:

- Deze methode kan niet worden gekwantificeerd
- Het is moeilijk om de factoren te documenteren die door de deskundigen of de deskundigengroep worden gebruikt

### **Program Evaluation and Review Technique / Critical Path Method (PERT/CPM)**

Program Evaluation and Review Technique (PERT) richt zich op het schatten van activiteiten en het bepalen van de afhankelijkheden. PERT vertegenwoordigt het project als logisch netwerk van activiteiten en hun bijbehorende voorganger en opvolger [SoftDev 95]. De bepalende factor voor PERT is tijd. Er worden drie verschillende tijdschattingen gemaakt voor een activiteit: optimistische, te verwachten en pessimistische schatting. De drie schattingen geven inzicht in de onzekerheid van de totale schatting. Door de verschillende schattingen kunnen meerdere scenario's gecreëerd worden, zoals het op of voor tijd eindigen van een activiteit. Dit geeft projectleiders inzicht in de gevolgen van uitgelopen activiteiten beter te begrijpen, inzicht in de gevolgen van een aangepaste planning in een verscheidenheid van manieren en voorzien van een betere planning in het begin van het project.

De drie schattingen die worden voorgeschreven door PERT worden door Imtech ICT Technical Systems (TS) tegen gesproken. In een interview met T. van der Put van Imtech ICT TS werd duidelijk dat, uit praktijk ervaring, het maken van een schatting met drie waarden de focus ligt op het schatten van drie waarden in plaats van het schatten van realistische waarden. Wanneer er gevraagd wordt twee waarden te schatten dan komen deze waarden meer overeen met de realiteit dan bij drie waarden.

Vanuit de afweging tussen projectdoorlooptijd en projectkosten is Critical Path Method (CPM) ontstaan. CPM bepaalt de projectduur die de som van directe en indirecte kosten minimaliseert. De beperkingen van het project worden in acht genomen en de opeenvolging(en) van activiteiten die de langste duur vereisen bepalen de kritieke weg(en) van het project. Het bepalen van het kritieke pad wordt vaak in combinatie met PERT gebruikt. PERT optimaliseert de projectduur en CPM minimaliseert de projectkosten.

De voordelen:

- Inzicht gevolgen van uitgelopen activiteit binnen projectplanning
- Beter begrijpen van projectplanning
- Betere planning in het begin van het project

De nadelen:

- Drie schattingen focust op het verkrijgen van drie verschillende schattingen en niet op realistische waarden

### **Work Breakdown Structure (WBS)**

De ontwikkeling van een projectplan berust op het hebben van een duidelijk en gedetailleerd inzicht in de taken, de geschatte tijdsduur die elke taak zal vergen, de afhankelijkheden tussen de taken en de opeenvolging waarin die taken moeten worden uitgevoerd. Waarna

beschikbaarheid moet worden bepaald om elke taak of groep taken aan de juiste werknemer toe te wijzen. Een mogelijke methode hiervoor is de Work Breakdown Structure (WBS).

De analysestructuur van WBS is een hiërarchische decompositie van een project of belangrijke activiteiten in opeenvolgende niveaus, waarbij elk niveau een fijnere analyse van voorafgaand niveau is. In definitieve vorm is een WBS zeer gelijkaardig in structuur en lay-out aan een documentoverzicht. Het detailleren van de niveaus in taken wordt herhaald tot een handelbare taakgrootte bereikt is. Een handelbare taak is één taak waarvan de verwachte resultaten gemakkelijk kunnen worden geïdentificeerd:

- Succes, mislukking, of de voltooiing van de taak kan gemakkelijk worden nagegaan.
- De tijd om de taak te voltooien kan gemakkelijk worden geschat.
- De resource vereiste van de taak kunnen eenvoudig worden bepaald.

Wanneer van elke taak een beeld is gevormd voor risico's, kosten en tijd, dan kan er aan de hand van een bottom-up methode de kosten voor het totale project worden bepaald. Een belangrijk overzicht vanuit de WBS is een decompositie van product en proces. [Boehm 00] geeft nog een ander voordeel van WBS aan. Door de decompositie wordt er een totaal overzicht gemaakt van kosten en plaats van de kosten binnen een project. Door deze informatie samen te vatten ontstaat een management budget controle instrument. Door deze informatie gedurende een periode vast te leggen wordt een database met de afspiegeling van de softwarekosten opgebouwd. Deze kan gebruikt worden voor een kostenmethode in de toekomst.

De voordelen:

- Verzekert dat de kosten voor activiteiten, zoals integratie en configuratie, niet worden vergeten binnen de schatting

### 3.5 Learning-oriented methoden

De learning-oriented typering erkent verschillende methoden, zoals case-based reasoning en neural networks. Voor de beeldvorming van de learning-oriented typering wordt in dit hoofdstuk Case-Based Reasoning (CBR) besproken.

#### Case-based reasoning (CBR)

Case-based Reasoning is een methode die gebruikt maakt van 'gevalse-specifieke' domeinkennis in plaats van 'generieke' kennis. Dit betekent dat om nieuwe problemen op te lossen CBR gebruik maakt van de kennis uit concrete problemen en situaties (cases) die zich al eerder voorgedaan hebben. Een CBR-systeem probeert 'oude' cases te vinden die vergelijkbaar zijn met het om handen zijnde probleem en vervolgens deze case (en de daarbij behorende oplossing) te gebruiken bij het oplossen van de nieuwe case. Een belangrijke eigenschap van CBR is dat het leert van nieuwe cases die het aangeboden krijgt en heeft opgelost zodat ook in de toekomst ditzelfde probleem snel opgelost kan worden.

De voordelen:

- Efficiënt (hergebruik)
- Bruikbaar in slecht doorgrondde domeinen

De nadelen:

- Slechte evaluatie leidt tot case-base vervuiling
- Veel / consistente cases nodig
- Case-based oplossingen zijn niet altijd optimaal

### 3.6 Meeteenheden voor kostenschatten

Ondanks de keuze voor een specifieke methode voor kostenschatten, zijn eenheden voor het schatten nodig. Er zijn twee soorten eenheden die onderscheiden worden: inspanning en omvang.

#### Inspanning

Inspanning is de grootte voor het aantal uur dat aan een softwarebouwproject wordt besteed. Inspanning wordt in de huidige situatie van Imtech ICT IT aangeduid met manuren. Deze eenheid is gekozen, omdat het hanteerbare eenheden zijn die overeenkomen met de duur van activiteiten van de ontwikkelaar. Bij grote projecten wordt er soms gekozen voor mandagen. Een mandag (acht manuren) is een grotere eenheid dan een manuur. Het nut van het schatten in manuren verdwijnt als de kleinste activiteit acht manuur (één mandag) beslaat. Om consistentie te behouden moeten toekomstige schattingen in manuren worden gearhiveerd. Consistent gebruik van één eenheid voorkomt verwarring en maakt vergelijkingen van schattingen onderling mogelijk.

#### Omvang

Om een indicatie van de omvang of te wel de grootte van een systeem aan te geven wordt in de huidige situatie van Imtech ICT IT dezelfde eenheid als inspanning gebruikt. In verschillende technische artikelen worden meerdere eenheden erkend. De hoofdgedachte achter het meten van de omvang is te verdelen in oplossingsgeoriënteerde (Source Line of Code, SLOC) en probleemgeoriënteerde (functionele omvang) omvang. Aan SLOC zijn veel nadelen verbonden. De definitie voor het tellen van één coderegel is niet vastgelegd in een standaard, waardoor verschillende uitkomsten voor een codebestand mogelijk zijn. De programmeertaal en programmeerstijl van een ontwikkelaar dragen bij aan het onmogelijk maken van het uniform tellen van de omvang in SLOC. Het grootste nadeel voor het schatten in SLOC ligt in het schatten van de omvang zonder één coderegel geprogrammeerd te hebben.

Om deze problemen te overbruggen is het oplossingsgeoriënteerde meten geïntroduceerd. Deze aanpak richt zich op het meten van de grootte van de problemen die verholpen moeten worden. Functie Punten (FP) is een voorbeeld van het meten van de omvang aan de hand van de systeemvereisten die de eindgebruiker heeft opgesteld. Een veelgebruikte methode die gebruik maakt van FP is de functiepuntenanalyse (FPA) [Nesma]. FPA definieert vijf verschillende functietypen die in een vroegtijdig stadium van ontwikkelen vanuit documentatie identificeerbaar zijn. In combinatie met de complexiteit van de functie geeft dit een indicatie van de omvang, zie tabel 8: functiepuntenmatrix. De vijf verschillende functietypen die gebruikt worden zijn: Logische Interne Gegevens Verzameling (ILGV), Gekoppelde Gegevens Verzameling (KGV), Invoer functie (IF), Uitvoer functie (UF) en Opvraag functie (OF), zie tabel 7 voor een beschrijving.

Na het berekenen van de functionele omvang wordt deze aangepast voor project technische situatie door een vermenigvuldigingsfactor. Deze aanpassing heeft zijn beperkingen, want voorgeschreven FPA-factoren met bijbehorend gewicht zijn niet van toepassing op alle softwarebouwprojecten. Een ander nadeel is de beperking van het meten van de complexiteit van logische bewerkingen of transacties.

Eerste reden voor het kiezen van FP is dat deze techniek eenvoudig toe te passen is. Een tweede reden is het eenduidig beschrijven [Nesma] van de procedure. Dit verhoogt de objectiviteit. Derde reden is de grote overeenkomst van FPA met de huidige manier van schatten. Het verdelen van het systeem in entiteiten (tabellen, objecten, schermen, transacties, etc.) en per complexiteit wordt een waarde toegekend (simpel, moeilijk, moeilijkst). In de volgende stap wordt per complexiteit de hoeveelheid manuren bepaald. De laatste stap is door middel van een optelling, rekeninghoudend met de complexiteit per functie, het totaal bepalen. Het schatten aan de hand van entiteiten, heeft veel overeenkomsten met FPA. Hierdoor zal de acceptatie van deze manier van schatten groter zijn en is de techniek makkelijker in te zetten.

## 4 Voorstel framework

Dit hoofdstuk gaat in op voorstel voor het nieuwe van het framework voor het schatten van softwarebouwprojecten bij Imtech ICT IT. Het framework wordt uiteen gezet aan de hand van de 'wat' en 'hoe' van het framework. In de laatste paragraaf worden de verwachtingen, waar Imtech ICT IT voor het slagen van het framework aan moet voldoen, besproken.

### 4.1 Wat is het framework

In de huidige situatie van Imtech ICT IT ontbreekt een duidelijke structuur voor het schatten van softwarebouwprojecten. Dit is onder andere de oorzaak van de problemen opgesomd in hoofdstuk 1. Het framework brengt structuur aan in het schattingsproces en voorziet het van een objectieve, controleerbare standaard voor het maken van kostenschattingen voor softwarebouwprojecten. Het framework is te karakteriseren als:

- Een standaard
- Een communicatiemiddel
- Transparant

Een standaard is een norm of model en bestaat uit voorgeschreven regels. De voorgeschreven regels samen vormen het framework. De voordelen van een framework zijn het gebruik maken van een gedeeld jargon en een eenduidige set afspraken voor het schatten van softwarebouwprojecten. De voorgeschreven processtappen geven structuur aan het schattingsproces. Tijdens het volgen van de processtappen wordt een schatting gemaakt voor de softwarebouw. Dit wordt aan de hand van een formulier gedocumenteerd. Het formulier maakt het mogelijk de schatting uniform vast te leggen en op later moment te vergelijken met eerder uitgevoerde schattingen.

In hoofdstuk 2, Projecten en kosten bij Imtech ICT IT, in wordt ingegaan op de kostenfactoren die een rol spelen bij het maken van een schatting. De verschillende kostenmethoden in hoofdstuk 3 houden rekening met de invloeden van kostenfactoren op de softwarebouw. Om een alle kostenomvattende schatting te maken is het van belang te weten welke kostenfactoren invloed hebben op de softwarebouwprojecten. De gegevens van eerder uitgevoerde kostenschattingen en bijbehorende gemaakte projectkosten zijn niet gedetailleerd genoeg voor het bepalen van de kostenfactoren. Het inzetten van de deskundigheid van de schatters is een oplossing voor dit probleem, maar introduceert een ander probleem. De expertise van de schatters is voornamelijk impliciete kennis. Deze impliciete kennis moet omgezet worden naar expliciete kennis. Om dit te bereiken moeten kostenfactoren die invloed uitoefenen op de projectkosten bespreekbaar worden gemaakt. Het inzetten van het

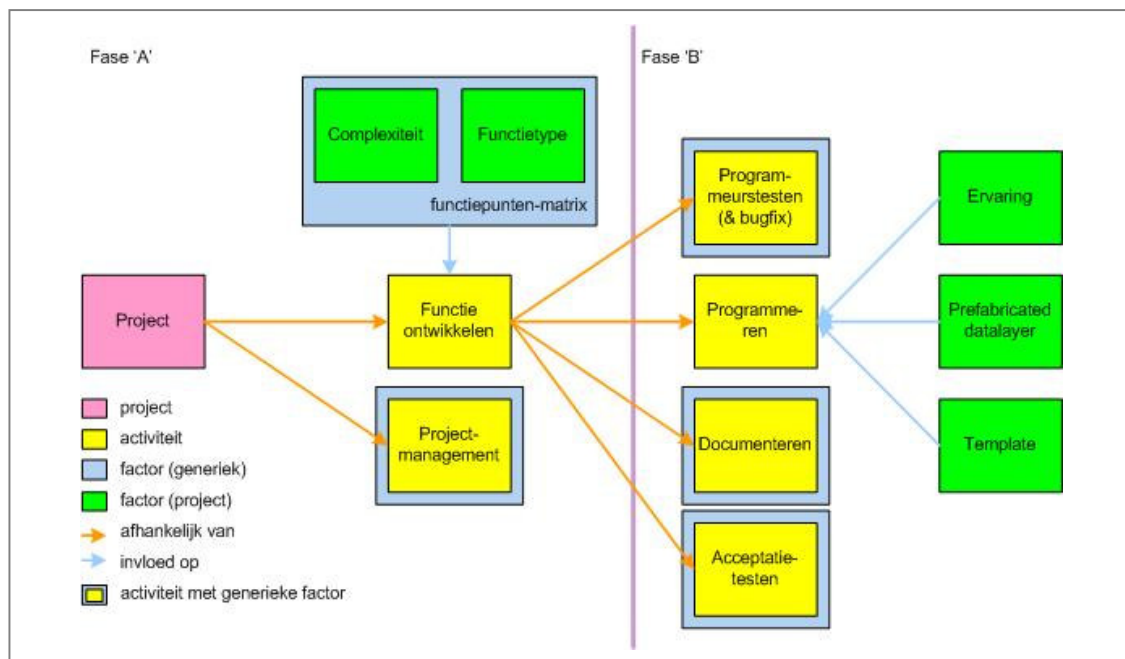
framework als communicatiemiddel maakt het mogelijk de kostenfactoren inzichtelijk te maken. Het bespreekbaar maken van de schattingen zorgt ervoor dat schatter aan het denken wordt gezet en bewust wordt gemaakt van de kostenfactoren, risico's en aannamen.

Het expliciet maken van de gedachten van de denkwijze van de schatter dient als invoer voor het kalibreren van het framework. Het framework wordt zodanig opgesteld dat het met nieuwe gegevens van projecten gekalibreerd kan worden. Het is kalibreren van het framework verhoogt de accuraatheid van de schattingen. Het framework wordt afgestemd op de softwarebouwprojecten van Imtech ICT IT. De gegevens die nodig zijn voor het kalibreren van het framework zijn afkomstig uit hetzelfde framework (learning-oriented).

Het framework moet een schatting controleerbaar, inzichtelijk en herleidbaar maken. Samenvattend het framework moet transparant zijn. Transparantie wordt verkregen door het documenteren van het schattingsproces en de algoritmen die het framework bevat. Een schatting is naderhand te herleiden aan de hand van de gedocumenteerde processtappen en bijbehorende kostenfactoren. Een van de belangrijkste kostenfactoren is de omvang. Hiermee wordt de grootte van het softwarebouwproject aangegeven. In de huidige situatie wordt de eenheid manuren gebruikt voor de weergave van de omvang. Om transparantie te verbeteren en een objectiviteit te verhogen biedt de eenheid functiepunten een concreter alternatief.

#### 4.2 Opzet van het framework

In het framework worden de verschillende factoren, die kosten bepalen bij Imtech ICT IT, als gewicht gebruikt in de kostenschatting. In hoofdstuk 2 worden de kostenbepalende factoren in kaart gebracht. Tussen de factoren zijn verschillende afhankelijkheden. De afhankelijkheden zijn in onderstaand figuur in kaart gebracht.



Figuur 3: activiteiten & afhankelijkheden

Een project bestaat uit verschillende fasen (figuur 1) waar een schatting voor gemaakt moet worden. De schatting voor een fase is afhankelijk van activiteiten die binnen de fase



uitgevoerd worden. In figuur 3 is een project afhankelijk van het ontwikkelen van een functies en projectmanagement. De activiteit projectmanagement valt buiten de scope van de onderzoeksvraag en wordt niet meegenomen binnen het framework. Deze en andere activiteiten zijn op een later moment wel toe te voegen aan het framework.

### Generieke en projectfactoren

De beeldvorming van het framework, in hoofdstuk 4.1, geeft aan dat een belangrijk onderdeel van het framework door de kostenfactoren wordt ingenomen. De kostenfactoren zijn te verdelen in generieke en project factoren. De keuze voor twee soorten factoren is voortgekomen uit de verschillende fasen van een project en hoe deze fasen beïnvloed worden door kostenfactoren.

De fasen, zoals deze weergegeven worden in figuur 1, vormen een onderdeel van het softwarebouwproject. In [Nageswaran 01], [Wiegiers 94], [Veen 97] en andere technische artikelen worden procentuele verdelingen verhoudingen voor de projectfasen beschreven. De projectfasen worden als percentages afgezet tegen de totale doorlooptijd van een softwarebouwproject. Percentages zijn corrigeer- en werkbaar, hierdoor zijn zij binnen het framework te plaatsen als eenheden voor de generieke factoren. In overeenstemming met Imtech ICT IT is er gekozen voor de waarden in tabel 4.

| <i>Activiteiten</i> | <i>Waarden</i> | <i>Beschrijving</i>  |
|---------------------|----------------|--|
| Programmeurtesten   | 5%             | Testen die uitgevoerd worden voor het ontwikkelen van een functie tijdens programmeren |
| Documenteren        | 10%            | Schrijven van de documentatie  |
| Acceptatietesten    | 15%            | De testen die uitgevoerd worden door de gebruiker voor acceptatie van de applicatie    |

Tabel 4: Generieke factoren voor softwarebouw

De generieke factoren zijn niet projectspecifiek en worden eenmalig bepaald en daarna afgestemd. Wanneer zij een accurate waarde aannemen zijn zij toepasbaar voor alle projecten van Imtech ICT IT.

Projectfactoren worden per functie voor elk project opnieuw bepaald. De projectfactoren hebben invloed op de omvang van een functie. Een verschil met de generieke factoren is de waarde van een projectfactor kan bestaan uit een percentage of functiepunten. In tabel 5 worden de verschillende projectfactoren en hun keuzes met bijbehorende waarden weergegeven.

| <i>Factor</i>           | <i>Keuzes</i>   | <i>Waarden</i>                             | <i>Beschrijving</i>  |
|-------------------------|---|--|--|
| Complexiteit            | Eenvoudig<br>Gemiddeld<br>Moeilijk                              | Gecombineerd met functietype <sup>1</sup>  | Indicatie van complexiteit aan de hand van aantal attributen en verwijzingen naar entiteiten   |
| Functietype             | ILGV <sup>2</sup><br>KGV<br>IF<br>UF<br>OF                      | Gecombineerd met complexiteit <sup>1</sup> | Geeft type van de functie weer. Gegevensverzameling beheren (ILGV) of raadplegen (KGV). Soort functie op scherm: invoer, uitvoer of opvragen.  |
| Ervaring                | Geen (0 – 1 jaar)<br>Weinig (1 – 5 jaar)<br>Veel (5 - ... jaar) | 0%<br>10%<br>15%                           | De programmeerervaring, Imtech ICT IT breed, met de betreffende functie  |
| Template                | Nee<br>Ja (40% hergebruik)<br><br>Ja (80% hergebruik)           | 0%<br>20%<br><br>40%                       | Geen gebruik van template<br>De structuur van een andere applicatie kan hergebruikt worden<br>Het kopiëren en plakken van code vanuit een andere applicatie is mogelijk                |
| Prefabricated datalayer | ...<br><br>Ja<br><br>Nee  | 0%<br><br>0%<br><br>20%                    | Er bestaat geen keuze tussen wel/ geen gebruik bestaande datalayer<br>Ja, gebruik maken van bestaande of gegenereerde datalayer<br>Nee, datalayer moet ontwikkeld worden vanaf scratch |

Tabel 5: Projectfactoren

De factoren complexiteit en functiepunttype komen voort uit FPA en bepalen de niet gecorrigeerde omvang van een functie, zie verderop in dit hoofdstuk. De factoren ervaring, template en prefabricated datalayer zijn vanuit de huidige situatie bij Imtech ICT IT vastgesteld als tijdsbepalende factoren voor een softwarebouwproject. De factoren hebben invloed op een activiteit, dat resulteert in een langere of kortere duur van de activiteit. De keuzemogelijkheden en waarden voor de factoren zijn op basis van ervaring en praktijk bepaald. Zij worden gekalibreerd, wanneer het framework in gebruik wordt genomen.

#### **Impact generieke en projectfactoren**

De percentages zoals deze hierboven worden weergegeven hebben een waarde nodig waarover zij berekend kunnen worden. Zoals in alle kostenmethoden wordt aangehaald dient de omvang van een functie de belangrijkste kostenfactor. De omvang van een functie maakt het mogelijk deze om te zetten naar een hoeveelheid manuren voor de softwarebouw.

De FPA maakt verschil in twee omvangfactoren, te weten de niet gecorrigeerde en gecorrigeerde omvang. De verschillen in de twee factoren hebben te maken met de organisatie specifieke eigenschappen voor softwarebouwprojecten. Door gebruik te maken complexiteit en functietype kan met een matrix (tabel 8) een omvang in functiepunten

<sup>1</sup> Zie functiepuntenmatrix, zie tabel 8

<sup>2</sup> ILGV (Interne Logische GegevensVerzameling), KGV (Koppelings GegevensVerzameling), IF (Invoer Functie), UF (Uitvoer Functie), OF (Opvraag Functie), zie tabel 7

toegekend worden. Echter zegt deze waarde maar voor een deel iets over de omvang van de functie. Wanneer alleen gebruik wordt gemaakt van deze waarde, dan zijn alle organisaties aan elkaar gelijk. Er is een aanpassing nodig voor de omvang, die de omvang weergeeft zoals deze binnen Imtech ICT IT bekend is. De gecorrigeerde omvang houdt rekening met de generieke en projectspecifieke factoren.

*Voorbeeld: een functie is moeilijk en heeft als functietype invoerfunctie. Dan is het aantal niet gecorrigeerde functiepunten 6.*

De gecorrigeerde omvang van het voorbeeld houdt rekening met de projectfactoren. De factoren programmeurtesten, documenteren en acceptatietesten zijn net als de generieke factoren percentages van de niet gecorrigeerde omvang. Deze berekeningen staan in tabel 6.

| Activiteit        | Invoer                | Factor                             | Formule  |
|-------------------|-----------------------|------------------------------------|--|
| Programmeurtesten | Niet gecorrigeerde FP | ProgTestPerc                       | Niet gecor. FP x ProgTestPerc  |
| Programmeren      | Niet gecorrigeerde FP | ErvPerc<br>PreFabPerc<br>TemplPerc | Niet gecor. FP + (Niet gecor. FP x (-ErvPerc + PreFabPerc + -TemplPerc)) |
| Documenteren      | Niet gecorrigeerde FP | DocPerc                            | Niet gecor. FP x DocPerc   |
| Acceptatietesten  | Niet gecorrigeerde FP | AccpTestPerc                       | Niet gecor. FP x AccpTestPerc  |

Tabel 6: Formules gecorrigeerde functiepunten

Voorbeeld: De functiegrootte in vorig voorbeeld was 6. Wanneer deze aangepast wordt voor Imtech ICT IT dan wordt de omvang van de functie bij weinig ervaring, geen 80% hergebruik en geen datalayer:

|                                   |              |
|-----------------------------------|--------------|
| <i>Niet gecorrigeerde omvang:</i> | 6            |
| <i>Ervaring:</i>                  | 0,6          |
| <i>Template:</i>                  | <u>2,2</u> - |
| <i>Datalayer:</i>                 | <u>1,1</u> + |
| <i>Totaal:</i>                    | 4,3 fp       |

Wanneer de generieke factoren worden meegenomen in de omvang wordt de gecorrigeerde omvang berekend. Hierbij wordt de 4,3 fp als basis genomen.

|                              |              |
|------------------------------|--------------|
| <i>Totaal:</i>               | 4,3 fp       |
| <i>Programmeurtesten</i>     | 0,2          |
| <i>Documenteren</i>          | 0,4          |
| <i>Acceptatietesten</i>      | <u>0,6</u> + |
| <i>Gecorrigeerde omvang:</i> | 5,59 fp      |

Door de functiepunten van de activiteiten afzonderlijk op te tellen wordt het totaal aantal functiepunten berekend. Dit wordt de gecorrigeerde functiepunten genoemd en geeft de werkelijke omvang van de functie weer. Het totaal van alle functies in de softwarebouwfase levert de totale omvang op.

De laatste stap voor het bereken van de manuren voor softwarebouw is de vermenigvuldiging met de norm. De norm is een sleutel (functiepunten/uur) die het aantal

functiepunten omzet naar manuren. De waarde voor de norm moet nog bepaald worden. Dit is vanwege het ontbreken van een dataset uit het verleden. Dit wordt verderop besproken.

### Het schattingsproces

Een van de karakteristieken van het framework is de structuur waar het proces een onderdeel van is. Het proces schrijft de stappen voor het maken van een schatting voor. De stappen die doorlopen moeten worden voor een schatting zijn afgeleid van Wideband Delphi.

Er wordt een schattingsteam samengesteld bestaande uit een moderator en minimaal twee, voorkeur drie, schatters. Een schatting moet met minimaal twee schatters gemaakt worden, want dan is het mogelijk een vergelijking van twee schattingen en de aannamen en risico's te bespreken. De schatters beschikken voldoende deskundigheid, zoals deze beschreven is in de huidige situatie. Een van de schatters krijgt de rol van moderator toebedeeld en leidt het schattingsproces. In een gezamenlijke zitting krijgen de schatters de specificatie (functioneel en technisch ontwerp) en het schattingsformulier voor de individuele schatting. Tijdens de zitting wordt de specificatie besproken, zodat onduidelijkheden vermeden worden. Eventueel kan er verdeling van verschillende mijlpaalproducten worden gemaakt, zodat per mijlpaalproduct geschat kan worden.

De volgende stap is de anonieme individuele schatting. Iedere schatter vult het schattingsformulier in (tabel 9). Het invullen van het schattingsformulier vergt een aantal stappen:

1. Softwarebouwproject, volgens WBS, opdelen in hanteerbare activiteiten. De activiteiten worden onder elkaar in de kolom gebruikersfunctie genoteerd.
2. Per activiteit moeten twee soorten schattingen worden gemaakt: manuren en FPA
  - 2.1. Per activiteit moet een schatting voor manuren gemaakt worden. De schatting bestaat uit twee schattingen: verwachte en pessimistische schatting manuren. Het verschil tussen de twee schattingen zit in de aannamen en risico's die bij de activiteit behoren. Deze moeten apart worden genoteerd, zodat deze inzichtelijk en benoemd kunnen worden.
  - 2.2. Per activiteit moet verschillende kostenfactoren bepaald worden voor de FPA. De kostenfactoren zijn: template, prefabricated datalayer, complexiteit, ervaring en functiepuntentype.

Nadat de individuele schattingen zijn afgerond wordt een groepsvergadering uitgeroepen, waarin de individuele resultaten besproken worden. Iedere schatter geeft per mijlpaalproduct aan welke activiteiten hij heeft gekozen en welke aannamen, risico's en kostenfactoren ermee gepaard gaan. Ook wordt de totaalschatting van de verwachte en pessimistische manuren per mijlpaalproduct gegeven. Aan de hand van de individuele resultaten van andere schatters kan een schatter zijn individuele schatting bijstellen, totdat wordt voldaan aan een van de volgende criteria:

- Resultaten binnen een acceptabele grens vallen
- Schatters geen reden meer zien om hun schatting bij te stellen
- Er twee herstelschattingen zijn geweest (aanpassingen van individuele schatting)
- De beschikbare tijd op is

Het resultaat van het schattingsproces is een aantal individuele schattingen die inzicht geven in aannamen, risico's, kostenfactoren en totaalschattingen. Het is van belang deze gegevens

te archiveren om een dataset op te bouwen. De dataset dient voor het kalibreren van het framework.

### **Het framework kalibreren**

In paragraaf Impact generieke en projectfactoren wordt de norm van het framework besproken. Deze waarde is nog niet bekend en moet nog bepaald worden. Door de komende tijd projecten volgens het voorgeschreven schattingsproces te laten verlopen, wordt een dataset opgebouwd. De dataset bevat softwarebouwproject gegevens die gebruikt worden voor het bepalen van de correctheid van de keuze kostenfactoren en het berekenen van de norm.

De kostenfactoren worden inzichtelijk door het bespreken van de kostenfactoren in groepszittingen. Elke schatting die gemaakt wordt, wordt aan de hand van de processtappen begeleid en levert risico's, aannamen en kostenfactoren op. Deze gegevens moeten door een persoon binnen de organisatie verzameld worden. Deze persoon is tevens verantwoordelijk voor het consistent gebruiken van het framework, want alleen dan is het te kalibreren. Fouten die in een schatting gemaakt worden moeten door deze persoon worden verwijderd, zodat dit geen ruis oplevert in het proces voor kalibreren.

Het berekenen van de norm is sterk afhankelijk van kostenfactoren. De juiste norm wordt bereikt, wanneer verschillende projecten die geschat zijn aan de hand van functiepunten worden omgezet naar een norm op basis van de werkelijk gemaakte uren voor het project. Het totaal aantal functiepunten moeten overeen stemmen met de gemaakte manuren. De verrekeningsleutel die tussen functiepunten en manuren geldt, moet bij deze projecten gelijk zijn. De afwijking van de norm bij de verschillende softwarebouwprojecten wordt bepaald door de kostenfactoren. Het afstemmen van de procentuele waarden van de kostenfactoren maakt het mogelijk de norm voor een softwarebouwproject aan te passen. Dit afstemmen is een proces dat constant uitgevoerd zal moeten worden, want zodra blijkt dan andere kostenfactoren niet in het framework zijn meegenomen, moeten deze toegevoegd worden. Dit zal impact hebben op de norm en het opnieuw kalibreren is onvermijdelijk.

## **5 Resultaten**

Deze scriptie geeft een antwoord op de onderzoeksvraag: Welke factoren spelen een rol bij het schatten van de kosten voor de technische realisatie van softwarebouwprojecten en hoe zijn deze te plaatsen in een passend framework voor Imtech ICT IT?

Het uitdiepen van de situatie bij Imtech ICT IT gaf al snel inzicht in het ontbreken van bruikbare gegevens uit het verleden. Imtech ICT IT baseert haar schattingen op de deskundigheid van de schatters. Het probleem wat hierbij optreedt is het gebruik van de impliciete kennis en subjectiviteit van een schatting. Een standaard ontbreekt en minimaliseert de controleerbaarheid en transparantie van een schatting.

Binnen de grote hoeveelheid kostenmethoden zijn twee soorten typering geschikt voor gebruik in het framework. De expertise-based typering en learning-oriented typering. De expertise-based typering brengt een structuur aan door de processtappen te beschrijven. Documenteren van de ondernomen stappen verhoogt de transparantie. Het proces wordt inzichtelijker. Het gebruik van het framework genereert een dataset, waarmee het framework gekalibreerd moet worden. Het kalibreren van de kostenfactoren, die vanuit de huidige situatie bij Imtech ICT IT zijn bepaald, verhoogt de accuraatheid.

Het framework voorziet in de behoefte aan een objectieve, controleerbare standaard. De standaard is vastgelegd en dus overdraagbaar voor nieuwe werknemers of te controleren door

een projectleider. Het framework vereist wel ervaring, maar specifieke ervaring op gebied van programmeren en hiervoor manuren schatten is geminimaliseerd.

## Bronvermeldingen

[Boehm 81] Boehm B., *Software Engineering Economics*, Prentice Hall PTR, ISBN:01-382-2122-7, 1981

[Boehm 00] Boehm B., Abts C., Chulani S., *Software development cost estimation approaches – survey*, Annals of Software Engineering, vol. 10, issue 1-4, pp. 177-205, ISSN:1022-7091, januari 2000

[Briand 98] Briand L., El Emam K., Surmann D., Wiecezorek I., Maxwell K., *An assessment and Comparison of Common Software Cost Modeling Techniques*, International Software Engineering Research Network Technical Report, ISERN-98-27, 1998

[Briand 99] Wiecezorek I., Briand L., Langley T., *A replicated Assessment and Comparison of Common Software Cost Modeling Techniques*, International Software Engineering Network Technical Report, ISERN-99-15, november 1999

[Briand 00] Briand L., Wiecezorek I., *Resource Estimation in Software Engineering*, Carleton University, Fraunhofer Institute for Experimental Software Engineering (IESE), ISERN-00-05, januari 2000

[Brooks 87] Brooks Frederick P., *No Silver Bullet: Essence and Accidents of Software Engineering*, Computer, Vol. 20, No. 4, April 1987, pp. 10-19.

[Cooper 99] Cooper A., *The Inmates Are Running the Asylum*, Macmillan Publishing Co. Inc., ISBN:06-723-1649-8, 1999

[Jorgensen 04] Jørgensen M., *A review of studies on expert estimation of software development effort*, Journal of Systems and Software, Vol 70, Issues 1-2, 2004, pp 37-60

[Kusters 99] Kusters R., Heemstra F., *Het schatten van softwarekosten - 10 jaar later*, Informatie, september 1999

[Leung 01] Leung H., Zhang F., *Software Cost Estimation*, The Pennsylvania State University CiteSeer Archives, maart 2001

[Levin 03] Levin V., *Cost Estimation for IT projects*, Management Concepts, augustus 2003

[Liming 97] Liming Wu, *The Comparison of the Software Cost Estimating Methods*, University of Calgary, maart 1997

[Meyer 91] Meyer M., Booker J., *Eliciting and analyzing expert judgement – a practical guide*, Knowledge Based Systems, vol. 5, Academic Press, 1991

[Nageswaran 01] Nageswaran S., *Test Effort Estimation Using Use Case Points*, Cognizant Technology Solutions, Quality week, San Fransisco, juni 2001

[Nesma] NEderlandse Software Metrieken Associatie, <http://www.nesma.nl/sectie/fpa/>

[Shepperd 97] Shepperd M., Schofield C., *Estimating Software Project Effort Using Analogies*, IEEE transactions on software engineering, vol. 23, no. 12, p. 736-743, november 1997

[SoftDev 95] Software Technology Support Center, *Report on Project Management and Software Cost Estimation Technologies*, april 1995, pp. 37-44

[Sommer 04] Sommerville I., *Software Engineering 7*, Chapter 26, Addison-Wesley, Pearson Education Ltd, 2004, pp. 612-640

[Standish 95] Standish Group International Inc., *The Standish Group Report - Chaos*, 1995

[Standish 01] Standish Group International Inc., *Extreme Chaos*, 2001

[Veen 97] Veenendaal E.P.W.M., Trienekens J.J.M., *Testen op basis van de kwaliteitsbehoeften van gebruikers*, Informatie, Jaargang 39, juli 1997

[Wiegers 94] Wiegers K. E., *Lessons from Software Work Effort Metrics*, Process Impact, Software Development Magazine, oktober 1994

## Verklarende woordenlijst

|                      |   |
|----------------------|---|
| Activiteit:          | Binnen een fase kunnen verschillende activiteiten plaatsvinden                |
| Classificatie:       | Een groep typeringen die gezamenlijk alle kostenschattingsmethoden bevatten   |
| Dataset:             | Verzameling gegevens over schattingen en resultaten van softwarebouwprojecten |
| Fase:                | Onderdeel van een softwarebouwproject   |
| Functiepuntentype:   | Geeft soort gebruikersfunctie weer: ILGV, KGV, IF, UF, OF (zie tabel 7)       |
| Gebruikersfunctie:   | Entiteit, scherm of functie die is beschreven in functioneel ontwerp.         |
| Kalibreren:          | Afstemmen van het framework aan de softwarebouwprojecten van Imtech ICT IT    |
| Kostenfactoren:      | Factoren die invloed hebben op de kosten voor een project of activiteit       |
| Manuren:             | Het aantal uren dat benodigd is voor de ontwikkeling van software             |
| Softwarebouwproject: | Project bij Imtech ICT IT voor de bouw van software                           |
| Typering:            | Een groep methoden die overeenkomstige eigenschappen vertonen                 |

## Bijlagen

### Functiepuntanalyse

Binnen de functiepuntanalyse zijn een aantal standaarden gedefinieerd. Deze zijn in onderstaande koppen en tabellen opgenomen.

#### Functiepunttypen

| <i>Functiepuntype</i>                        | <i>Beschrijving</i>   |
|--|---|
| Interne Logische Gegevens Verzameling (ILGV) | Een Interne Logische Gegevensverzameling bevat permanente, voor de gebruiker relevante gegevens. De gegevens worden door het systeem gebruikt en onderhouden. Onder "onderhouden" verstaat FPA het toevoegen, wijzigen of verwijderen van gegevens.   |
| Koppelings-Gegevensverzameling (KGV)         | Ook een Koppelings-Gegevensverzameling bevat permanente, voor de gebruiker relevante gegevens. Deze gegevens worden door het systeem gebruikt, maar worden door een ander systeem onderhouden (voor dat andere systeem is het dus een ILGV).  |
| Invoerfunctie (IF)                           | Een Invoerfunctie verwerkt gegevens in een ILGV van het systeem. Voorbeelden: het wijzigen van de gegevens van een klant in het klantenbestand; het inlezen van bestellingen in een ordersysteem; het medium is hierbij niet van belang: papier, scherm, cartridge, datacom, enzovoorts.  |
| Uitvoerfunctie (UF)                          | Een Uitvoerfunctie presenteert gegevens uit het systeem. Voorbeelden: het afdrukken van alle debiteuren; het aanmaken van facturen; het aanmaken van een diskette met betalingsopdrachten; het medium is hierbij niet van belang: papier, scherm, magneetband, datacom, enzovoorts.   |
| Opvraagfunctie (OF)                          | Een Opvraagfunctie is een speciaal (eenvoudig) soort uitvoerfunctie. Een opvraagfunctie presenteert gegevens uit het systeem op basis van een uniek identificerend zoekgegeven, waarbij geen aanvullende bewerkingen (zoals berekeningen of het bijwerken van een gegevensverzameling) plaats hebben.<br>Voorbeeld: Het tonen van de gegevens van de klant met klantnummer 123456789. |

Tabel 7: Functiepuntypen volgens [Nesma]

### Functiepuntenmatrix

| Type gebruikersfunctie                      | <i>Complexiteit</i> |           |          |
|---|---------------------|-----------|----------|
|   | Eenvoudig           | Gemiddeld | Moeilijk |
| Interne Logische Gegevensverzameling (ILGV) | 7                   | 10        | 15       |
| Koppelings Gegevensverzameling (KGV)        | 5                   | 7         | 10       |
| InvoerFunctie (IF)                          | 3                   | 4         | 6        |
| UitvoerFunctie (UF)                         | 4                   | 5         | 7        |
| OpvraagFunctie (OF)                         | 3                   | 4         | 6        |

Tabel 8: Functiepuntenmatrix volgens [Nesma]



## Schattingsformulier

### Schattingsformulier

|                               |              |
|-------------------------------|--------------|
| Naam                          | Xxx          |
| Naam project                  | Xxx          |
| Datum                         | 12 aug 2005  |
| Schattingseenheid voor werk   | Manuren      |
| Schattingseenheid voor omvang | Funciepunten |

| Gebruikersfunctie/<br>Entiteiten | Ontwikkelen |               | Template            | Prefabricated<br>DataLayer | Rules<br>Complexiteit | Ervaring            | Functietype |     |    |    |    |   |
|----------------------------------|-------------|---------------|---------------------|----------------------------|-----------------------|---------------------|-------------|-----|----|----|----|---|
|                                  | Verwachte   | Pessimistisch |                     |                            |                       |                     | ILGV        | KGV | IF | UF | OF |   |
| Xxx                              | 16          | 32            | Nee                 |                            | Moeilijk              | Nee (0 - 1 jaar)    |             | 1   |    |    |    | 1 |
| Xxx                              | 22          | 32            | Ja (40% hergebruik) |                            | Eenvoudig             | Weinig (1 - 5 jaar) |             |     |    |    | 1  |   |
| Xxx                              | 56          | 80            | Ja (40% hergebruik) |                            | Eenvoudig             | Weinig (1 - 5 jaar) |             |     | 3  |    |    | 1 |
| Xxx                              | 5           | 10            | Ja (40% hergebruik) |                            | Eenvoudig             | Weinig (1 - 5 jaar) |             |     |    |    |    | 1 |
| Xxx                              | 22          | 30            | Nee                 |                            | Gemiddeld             | Weinig (1 - 5 jaar) |             |     |    |    |    |   |
| Xxx                              | 10          | 16            | Ja (40% hergebruik) |                            | Eenvoudig             | Weinig (1 - 5 jaar) |             |     |    |    | 1  |   |
| Xxx                              | 15          | 24            | Ja (40% hergebruik) |                            | Eenvoudig             | Weinig (1 - 5 jaar) |             |     | 1  |    |    |   |
| Xxx                              | 15          | 24            | Ja (40% hergebruik) |                            | Eenvoudig             | Veel (5 - ... jaar) |             |     |    |    | 1  |   |
| Totaal ontwikkelen               | 161         | 248           |                     |                            |                       |                     |             |     |    |    |    |   |
| Documentatie                     | 16,1        |               |                     |                            |                       |                     |             |     |    |    |    |   |
| Projectmanagement                | 32,2        |               |                     |                            |                       |                     |             |     |    |    |    |   |
| Acceptatietesten                 | 24,15       |               |                     |                            |                       |                     |             |     |    |    |    |   |
| Totaal project                   | 233,45      |               |                     |                            |                       |                     |             |     |    |    |    |   |

Tabel 9: Schattingsformulier