

# VU Research Portal

## Bit-to-board analysis for IT decision making

Kwiatkowski, L.M.

2012

### **document version**

Publisher's PDF, also known as Version of record

[Link to publication in VU Research Portal](#)

### **citation for published version (APA)**

Kwiatkowski, L. M. (2012). *Bit-to-board analysis for IT decision making*.

### **General rights**

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal ?

### **Take down policy**

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

### **E-mail address:**

[vuresearchportal.ub@vu.nl](mailto:vuresearchportal.ub@vu.nl)

## CHAPTER 4

### Samenvatting

#### 4.1 Broncode: een goudmijn van management informatie

Het onderzoek dat in dit proefschrift wordt beschreven is uitgevoerd onder de auspiciën van het EQUITY project. EQUITY staat als afkorting voor *Exploring Quantifiable IT Yields* en het project wordt gesubsidieerd door de Nederlandse Organisatie voor Wetenschappelijk Onderzoek (ZWO) en het Ministerie van Economische Zaken uit gelden die worden beheerd door de JACQUARD stichting. De doelstelling van EQUITY is een kwantitatieve basis te leggen onder de besluitvorming over IT. Veel beslissingen over de inzet en het onderhoud van IT middelen in het bedrijf worden thans gevoelsmatig genomen en ontberen een rationele (kwantitatieve) grondslag. In het bedrijfsleven is men zich nauwelijks bewust van het feit dat veel essentiële kwantitatieve management informatie ligt opgesloten in de broncode van de informatiesystemen die de bedrijfsactiviteiten ondersteunen. In dit proefschrift wordt verslag gedaan van twee toepassingen van broncode analyse die gericht is op het genereren van management informatie. In beide gevallen is gebruikgemaakt van praktijkdata en hebben we de resultaten van de case studies ter toetsing voorgelegd aan praktijkfunctionarissen.

Het hoeft nauwelijks betoog dat IT een onmisbare productiefactor is voor de meeste bedrijven en overheidsinstellingen voor het realiseren van hun organisatiedoelstellingen. De bekende IT service provider Gartner meldt dat het totale bedrag dat wereldwijd door het bedrijfsleven aan IT werd besteed in 2008 circa 3,4 biljoen euro bedroeg. De grote afhankelijkheid van de business van IT ondersteuning heeft niet alleen vergaande kosten consequenties, maar brengt ook grote risico's met zich mee. Voor veel bedrijven geldt dat de continuïteit van de bedrijfsvoering ernstig in gevaar wordt gebracht indien bepaalde cruciale informatiesystemen langer dan een etmaal down gaan. Het onderstaande staatje geeft een goed beeld van de kostenimpact van een uur uitval van IT ondersteuning. Het overzicht is gemaakt door de Fiber Channel Industry Association en overgenomen uit [153]. Voor een aantal verschillende industrieën en bedrijfsactiviteiten is geschat hoe groot de

financiële schade is indien de ondersteunende IT systemen gedurende 1 uur niet beschikbaar zijn als gevolg van een hardware- of softwarestoring.

Business activity	Industry	Hourly downtime costs
brokerage operations	finance	\$6,450,000
CC authorizations	finance	\$2,600,000
pay-per-view	media	\$150,000
home shopping (TV)	retail	\$113,000
catalog sales	retail	\$90,000
airline reservations	transportation	\$90,000
tele-ticket sales	media	\$69,000
package shipping	transportation	\$28,000
ATM fees	finance	\$14,500

Table 4.1: Information systems hourly downtime cost per business activity.

Zie dat de verliezen in sommige gevallen kunnen oplopen tot zes en een half miljoen dollar per uur! Indien men zich realiseert dat de IT systemen met grote regelmaat moeten worden aangepast en uitgebreid vanwege de steeds wisselende eisen die de omgeving (markt) aan de business stelt, dan is duidelijk dat een goed inzicht in de kosten, de kwaliteit en het aanpassingsvermogen van de bestaande IT- portfolio onontbeerlijk is. In dit proefschrift laten we zien dat de broncode van de informatiesystemen een schat aan informatie bevat die na een paar bewerkingsslagen aan het management kan worden aangeboden als basis voor besluitvorming. We laten zien hoe relevante feiten uit de broncode kunnen worden opgediept door analyse van compiler uitdraaien van recente productieversies van de informatiesystemen. Broncode analyse is op zich niet nieuw en kan bogen op een rijke ervaringsgeschiedenis. Voor onze doeleinden hebben we dan ook gebruikgemaakt van een aantal bekende geautomatiseerde hulpmiddelen voor broncode analyse. Onder meer hebben we gebruikgemaakt van gereedschap dat gebaseerd is op de zogenaamde lexicale analyse methode. Teneinde de analyse eenvoudig te houden en niet duurder te maken dan nodig hebben we in een aantal gevallen zelf analyse software geschreven. Deze kwam in de plaats van op de markt verkrijgbaar gereedschap dat voor onze analyse doeleinden een te zwaar en te duur hulpmiddel zou zijn geweest. Bij de uitvoering van de broncode analyse is het motto steeds geweest: maak het niet moeilijker en duurder dan nodig en volg een werkwijze die door andere bedrijven gemakkelijk te kopiëren is.

Het grote voordeel van broncode als informatiebron is dat het een zuivere informatiebron is. Zuiver in de zin dat de feiten die aan het licht worden gebracht niet kunnen zijn gemanipuleerd. Dit in tegenstelling tot andere informatiebronnen die bestaan uit gegevens die door medewerkers zijn verzameld en ingevoerd. Bij het vastleggen van die gegevens kunnen persoonlijke waardeoordelen een belangrijke rol spelen. Vaak zijn de bestanden bovendien niet goed bijgehouden en daardoor onvolledig. Denk bijvoorbeeld aan documentatie betreffende functioneel onderhoud en uitbreiding van functionaliteit van systemen. Ook beschikken veel organisaties vaak niet over gegevens betreffende kosten en productiviteit bij systeemontwikkeling en onderhoud. Indien beschikbaar, dan is men vaak pas laat begonnen met het meten en vastleggen van dergelijke projectgegevens en is

tijdreeks analyse daardoor niet mogelijk. Voor informatiesystemen die lang geleden zijn gebouwd en een aantal keren zijn aangepast geldt vaak dat de systeemdokumentatie verloren is gegaan en de programmeurs die de bouw en onderhoudswerkzaamheden hebben verricht de dienst reeds lang hebben verlaten. Voor de broncode gelden deze bezwaren niet. De laatst bijgewerkte versie van de broncode is per definitie beschikbaar en vaak kan uit de commentaarregels die de programmeurs tijdens de bouw- en onderhoudswerkzaamheden aan de broncode hebben toegevoegd de historie van het systeemonderhoud voor het overgrote deel worden gereconstrueerd.

Zoals gezegd, doet dit proefschrift verslag van twee toepassingen van broncode analyse ten behoeve van het genereren van management informatie. In beide gevallen is gebruikgemaakt van praktijkdata en zijn de resultaten ter toetsing voorgelegd aan praktijkfunctionarissen. De praktijkdata is afkomstig van een internationaal opererende financiële instelling en bestaat uit de broncode van de informatiesystemen van een van haar grote business units. In hoofdstuk 3 wordt beschreven hoe we de plausibiliteit van de data hebben gecheckt door een aantal controles op de data uit te voeren. De testresultaten waren alle positief. We geven nu een aantal karakteristieken van de IT portfolio. De betreffende portfolio bestaat uit 47 informatiesystemen die in COBOL zijn geschreven en draaien op een mainframe. De 47 systemen bestaan in totaal uit 8198 modules (COBOL programmas). De code is voor ca 40% handmatig geschreven en voor het overige deel gegenereerd door COBOL codegeneratoren, zoals TELON, COOL:GEN en CANMAN. De totale code van de IT portfolio bestaat uit circa 18 miljoen regels. Voorwaar een astronomisch groot aantal! Het is duidelijk dat het niet doenlijk is om 18 miljoen regels code handmatig te analyseren, maar dat daarvoor geautomatiseerde hulpmiddelen nodig zijn. De oudste portfolio systemen stammen uit de jaren zestig van de vorige eeuw en de meest recente systemen zijn van 2009. De portfolio bestrijkt dus een tijdsperiode van vijftig jaar. In die periode hebben zich tal van ontwikkelingen op IT gebied voorgedaan die hun sporen in de broncode hebben achtergelaten. Zo heeft de programmeertaal COBOL een evolutie doorgemaakt. Als gevolg daarvan zijn vele verouderde, inefficiënte programmeerconstructies in de code te vinden. Ook de codegeneratoren hebben een ontwikkeling in de tijd doorgemaakt en de code die is gegenereerd door de ene codegenerator, verschilt van de andere. Een probleem is dat geen van de codegeneratoren kan worden afgestoten, zolang de portfolio nog systemen bevat waarvan modules zijn gegenereerd door de codegenerator. De licentiekosten blijven doorlopen en de kennis over het gebruik van de codegenerator moet op peil worden gehouden. Vervanging van de gegenereerde code door handgeschreven code zou een oplossing zijn, maar is vaak een riskante en kostbare operatie, zeker als de software cruciale bedrijfsactiviteiten ondersteunt. Een ander probleem dat een gevolg is van de lange levensduur van de portfolio is de noodzaak verschillende COBOL compilers *up and running* te houden. Het besturingssysteem van het IBM mainframe is in de tijd sterk geëvolueerd. Van de meer dan acht duizend modules waaruit de informatiesystemen zijn opgebouwd kan 40% alleen worden gecompileerd met de verouderde IBM OS/VS COBOL compiler, die niet meer door IBM wordt ondersteund. Niet alleen is de programmeertaal COBOL in de loop van de tijd sterk verbeterd en zijn steeds meer geavanceerde COBOL code generatoren beschikbaar gekomen, maar ook hebben de database management systemen en transactie monitoren een ontwikkeling doorgemaakt in de afgelopen 50 jaar. Voor gegevensverwerkende bedrijven is de opslag en integriteit-

bewaking van data in databases, het bijwerken van data en het efficiënt kunnen opvragen van data uit databases een cruciale bedrijfsactiviteit. Voor het opvragen van gegevens uit de database wordt gebruikgemaakt van speciaal daarvoor ontworpen opvraagtaalen, zoals SQL, dat staat als afkorting voor Structured Query Language. Een efficiënt gebruik van SQL, gecombineerd met een efficiënte technische database infrastructuur, leidt tot grote kostenbesparingen en lage response tijden. Door verouderde en inefficiënt werkende SQL programmeerconstructies via broncode analyse op te sporen en vervolgens te optimaliseren, gebruikmakend van nieuwe inzichten en programmeerhulpmiddelen, kunnen grote besparingen worden gerealiseerd met een laag afbreukrisico voor de continuïteit van de bedrijfsvoering.

We geven nu een samenvatting van de twee case studies die we hebben uitgevoerd. Bij beide case studies staat het distilleren van (kwantitatieve) management informatie uit de broncode centraal.

## 4.2 Case studie 1: strategisch IT management

De eerste case studie betreft IT besturing op strategisch niveau (vaak aangeduid met IT Governance). Een specifieke taak van de ondernemingsleiding is het ontwikkelen en uitdragen van een strategische visie op het toekomstige reilen en zeilen van de onderneming. Een belangrijk onderdeel daarvan is de visie op de inzet van IT voor het realiseren van de bedrijfsdoelstellingen. Op strategisch niveau gaat het om de beantwoording van vragen als:

- Wat is de economische waarde van de bestaande portfolio van IT systemen? Voor welk bedrag zouden deze kapitaalgoederen op de balans moeten worden opgevoerd?
- Welke kosten zullen in de toekomst bij ongewijzigd beleid zijn gemoeid met het in stand en operationeel houden van de informatiesystemen?
- Wat is de kwaliteit van de bestaande IT systemen? Hoe groot is het risico dat de uitvoering van essentiële bedrijfsactiviteiten stopt als gevolg van falende IT systemen? Kan dat risico worden verlaagd door herbouw of vervanging van de informatiesystemen die aan het risico debet zijn?

Dit soort vragen is aan de orde bij strategische management beslissingen over IT middelen. De ondernemingsleiding moet aangeven welke activiteiten voorrang krijgen bij het onderhoud en beheer van haar IT middelen.

In hoofdstuk 2 wordt uiteengezet hoe door analyse van broncode een antwoord kan worden gegeven op bovengenoemde typen management vragen. Als eerste wordt de vraag behandeld hoe de economische waarde van de portfolio van informatiesystemen kan worden bepaald. Door te schatten hoeveel de herbouw van de informatiesystemen vandaag de dag zou kosten (gebruikmakend van de meest geëigende productiewijze) wordt een goede indicatie gekregen van de vervangingswaarde van de portfolio. Met behulp van broncode analyse en gebruikmakend van de zogenaamde *backfiring* methode [80] [82, p. 79] wordt eerst een schatting gemaakt van de omvang van de bestaande informatiesystemen,

gemeten in functiepunten. Een functiepunt is een door de industrie en wetenschap algemeen geaccepteerde synthetische maat voor het meten van de omvang en complexiteit van een systeem. Zoals in de bouwwereld bij het opstellen van begrotingen de kubieke meterprijs wordt gehanteerd, zo kan men bij het opstellen van een kostenbegroting van een systeem ontwikkelproject uitgaan van de kosten per functiepunt. Vervolgens worden publieke *benchmark* formules gebruikt om op basis van de geschatte omvang van de informatiesystemen te schatten hoeveel kosten, tijd en risico er mee gemoeid zouden zijn om de systemen met de beschikbare kennis en hulpmiddelen van nu te bouwen. De benchmark formules worden in hoofdstuk 2 uitvoerig toegelicht en beargumenteerd wordt waarom ze mogen worden toegepast in ons specifieke geval. Om een beeld te krijgen van de verwachte aangroei van de portfolio in de toekomst wordt gebruikgemaakt van ervaringscijfers uit het verleden. Van een groot aantal modulen kan uit analyse van de broncode het groeipad worden gedestilleerd dat is doorlopen sedert de datum van hun creatie. In de loop van de tijd is de functionaliteit van de modulen uitgebreid en dat heeft geresulteerd in een toename van het aantal functiepunten. Door extrapolatie van deze groeicijfers wordt een redelijk betrouwbare schatting verkregen van de aangroei van de portfolio als totaal bij ongewijzigd beleid. Een jaarlijkse toename van het aantal functiepunten op portfolio niveau met 8,7% ligt in de lijn der verwachtingen. De kosten repercussies hiervan in termen van toename van onderhoud- en exploitatiekosten worden met behulp van publieke *benchmark* formules ingeschat.

Risico analyse is een chapter apart. Zo blijkt uit broncode analyse dat 1237 modulen (ca 35% van alle handgeschreven modulen) het stempel complex verdient. Dat wil zeggen: voor 1237 modulen geldt dat de berekende McCabe indicator (een algemeen geaccepteerde maat om de complexiteit van software te meten [108]) meer dan 50 bedraagt, hetgeen volgens ervaring een niet te verwaarlozen risicofactor is bij onderhoud en vernieuwbouw. Duidelijk is dat het gebruik van verouderde Cobol compilers en codegeneratoren, behalve hogere kosten (licenties), ook extra risico met zich meebrengt. Uit onze broncode analyse blijkt dat veertig procent van de portfolio modulen voor compilatie afhankelijk is van de verouderde, niet meer door IBM ondersteunde IBM OS/VS COBOL compiler. De kennis over deze verouderde hulpmiddelen is normaliter schaars aanwezig, omdat de medewerkers die ermee hebben gewerkt de dienst hebben verlaten. Verder is duidelijk dat systemen van oude datum, die voor een belangrijk deel uit modulen bestaan die zijn geschreven in niet meer gebruikte dialecten van COBOL, een groter risico van falen lopen dan systemen van recentere datum. Indien zulke systemen dan ook nog met grote regelmaat worden aangepast (hoge volatiliteit) en cruciale bedrijfsactiviteiten ondersteunen, kan worden gesproken worden van een hoog risico gebied (hot spot) binnen de portfolio. In hoofdstuk 2 wordt beschreven hoe zulke *hot spots* met behulp van broncode analyse in kaart kunnen worden gebracht en kunnen worden afgebakend.

Met behulp van *what-if* scenario analyse wordt een beeld verkregen van het risico profiel van de portfolio in de toekomst bij ongewijzigd beleid. Ook worden met behulp van *what-if* scenario analyse mogelijke migratietrajecten, gericht op kostenverlaging en risicoverlaging, geëvalueerd. Daarbij worden de systemen gerangschikt in volgorde van kritisch belang voor de bedrijfsvoering. Een systeem wordt als bedrijfskritisch aangemerkt, indien het systeem de uitvoering van een bedrijfsactiviteit met een hoog transactievolume ondersteunt. Bij migratietrajecten gericht op kosten- en risicoverlaging blijken

de topkritische bedrijfsapplicaties in belangrijke mate te worden geraakt.

### 4.3 Case studie 2: verlaging operationele kosten door MIPS management

MIPS staat als afkorting voor Million Instructions Per Second. De kosten die het Rekencentrum de eindgebruiker in rekening brengt voor het draaien van een applicatie zijn voor het overgrote deel gebaseerd op het MIPS-verbruik door de applicatie. Duidelijk is derhalve dat het vanuit een kostenooptpunt loont te trachten het verbruik van het aantal MIPS te minimaliseren. Hoofdstuk 3 laat zien dat broncode analyse een belangrijk hulpmiddel is bij het opsporen van inefficiënt geschreven code, die debet is aan een hoog MIPS verbruik. Feitelijk is het de combinatie van programmatuur en database infrastructuur die de inefficiëntie veroorzaakt, maar door de *verdachte* programmeerconstructies op te sporen in de code wordt tegelijkertijd de schijnwerper gezet op mogelijke technische inefficiënties.

In de tweede case studie wordt ingezoomd op SQL code die voor IMS (Information Management System) transacties is geschreven om data op te vragen uit DB2 databases. DB2 is een ruim verspreid relationeel gegevenssysteem dat wordt gebruikt in de productieomgeving (massale transactieverwerking) van gegevensverwerkende bedrijven. In DB2 en andere relationele systemen wordt SQL (Structured Query Language) gebruikt als interface tussen de gegevensbank en toepassingsprogrammatuur en als zelfstandige vraagtaal. Ons onderzoek wijst uit dat wekelijks bij de uitvoering van 72% - 80% van de belangrijkste IMS transacties een database wordt geraadpleegd. De MIPS consumptie van deze transacties blijkt gemiddeld een factor 2 hoger te liggen dan die van transacties die geen database raadplegen. De vraagtaal SQL kent veel uitdrukkingen, waarvan gebruik kan worden gemaakt bij het opslaan, opvragen of bijwerken van gegevens. Wij hebben bij ons onderzoek de schijnwerper gezet op het SELECT statement. Binnen SELECT zijn WHERE en FROM voorbeelden van veelgebruikte operaties. De programmering in SQL luistert bijzonder nauw. Inefficiënte programmeerconstructies leiden snel tot een excessief MIPS verbruik bij de transactieverwerking. Indien het om massale transactieverwerking gaat kunnen de kosten daardoor hoog oplopen. In hoofdstuk 3 wordt beschreven hoe met behulp van eenvoudig uit te voeren broncode analyse *verdachte* SQL programmeerconstructies kunnen worden opgespoord en aan experts ter beoordeling kunnen worden aangeboden. Bij de broncode analyse wordt onder meer gebruikgemaakt van gereedschap dat is gebaseerd op de zogenaamde lexicale analyse methode. Zoals eerder gezegd gaat het hier om een portfolio die in het totaal een kleine twintig miljoen aan code regels omvat. Zonder geautomatiseerde hulpmiddelen is het ondoenlijk een dergelijk groot bestand te analyseren. Uit de analyse blijkt dat 668 van de 23004 modules waaruit de informatiesystemen zijn opgebouwd SQL programmeerconstructies bevatten die mogelijk inefficiënt zijn geschreven. Een aantal van deze modules heeft grote invloed op het MIPS verbruik, omdat ze bij hun uitvoering andere modules aanroepen, die weer andere IMS transacties ondersteunen. Optimalisatie van de SQL code van deze modules heeft dus potentieel een veel groter MIPS consumptie besparingseffect dan optimalisatie van de SQL code van modules die weinig of geen relaties met andere modules hebben en de uitvoering van transacties ondersteunen met een laag transactievolume. In hoofdstuk 3 wordt beschreven

hoe we met behulp van broncode analyse de aanroep (call) relaties tussen de modulen in kaart hebben gebracht. Het resultaat is onder meer samengevat in de vorm van een graaf. De potentiële besparing op het MIPS verbruik door optimalisatie van de SQL code wordt nog veel groter indien alle betrokken IMS transacties een groot omzetvolume kennen. Uiteindelijk kon het zoekgebied worden verkleind tot 0,78% van de 23000 COBOL programmas. We hebben de resultaten van onze *quick scan* geëvalueerd door ze te vergelijken met de resultaten van een kleinschalig MIPS besparingsproject dat eerder door het bedrijf was uitgevoerd. Met dit project werd een kostenbesparing van ca 3,8% op de maandelijkse MIPS kosten rekening gerealiseerd. Het bleek, dat alle SQL code die het team had herschreven (geoptimaliseerd), in onze scan als potentieel inefficiënt geprogrammeerde SQL code was onderkend. We hebben een schatting gemaakt van de kostenbesparingen die het team zou hebben kunnen realiseren, indien het team alle door ons gevonden potentieel inefficiënte SQL code onder ogen zou hebben gekregen en waar nodig had geoptimaliseerd. Het team had dan 180 modulen bekeken en een kostenbesparing van om en nabij 7% op de maandelijkse MIPS kostenrekening gerealiseerd. De belangrijkste conclusie die uit de tweede case studie kan worden getrokken is, dat door middel van broncode analyse op goedkope en snelle wijze een inventarisatie kan worden gemaakt van de SQL code die mogelijk inefficiënt is geschreven en daarom in aanmerking komt voor optimalisatie onderzoek. De uiteindelijke beoordeling is aan de SQL experts, maar veel tijd en kosten worden bespaard door de inzet van deze dure en schaarse expertise te richten op de meest veelbelovende activiteiten, waarvan het hoogste kostenbesparingsrendement kan worden verwacht.



