

Het is een heldere zonnige, maar koude middag als ik het parkeerterrein van het luxueuze vijf sterrenhotel Karel V in Utrecht oprij. In de Brasserie van het hotel heb ik een afspraak met Ivar Jacobson, de oudste van de three amigos Booch, Jacobson en Rumbaugh. Jacobson is verantwoordelijk voor een belangrijk deel van de modelleertechnieken van de facto modelleertaal Unified Modeling Language (UML). Bovendien geldt de Zweed als de vader van de systeemontwikkelmethode Rational Unified Process. Tijd voor een gesprek in twee delen over systeemontwikkelprocessen en de agilititeit van RUP en over het verleden en de toekomst van modelleren, ontwerpen en programmeren.

terview

# Drie procent kans de wereld te verbeteren

## *Ivar Jacobson over Objectory, RUP en agility*

Even later neem ik plaats bij de open haard in de Brasserie van het hotel. Aan de andere kant van de ruimte praat Jacobson enthousiast met een journalist die waarschijnlijk voor het eerst in zijn leven iets schrijft over UML. Jacobson blijft rustig en legt geduldig de principes van de modelleertaal uit, daarbij accentuerend dat de modelleertaal geen programmeertaal is. Voor de zoveelste keer in zijn lange loopbaan.

**GRIJZENDE ZWEED** Ivar Jacobson begint zijn carrière in Zweden bij telecombedrijf Ericsson, waar hij bijvoorbeeld al in 1976 de termen component en inter-

mede gebaseerd op de jarenlange ervaringen bij Ericsson”, vertelt Jacobson enthousiast over deze kleurrijke periode.

“In die tijd waren er eigenlijk geen systeemontwikkelprocessen voor objectgeoriënteerde projecten. Dat was een groot gemis”, beschrijft hij. “Het Objectory Process was het eerste systeemontwikkelproces in dit vakgebied.” Glimlachend verhaalt Ivar Jacobson over zijn eerste bezoek aan de OOPSLA in 1987, de grote jaarlijkse wereldwijde conferentie over objectoriëntatie. “Tijdens deze conferentie gaf ik een eerste presentatie over ons systeemontwikkelproces. De reacties waren overweldigend. Zonder uitzondering uitermate positieve reacties. Als gevolg van deze presentatie werd Objectory meteen al benaderd om het proces ook in de Verenigde Staten bij klanten te implementeren.”

Bij thuiskomst in Zweden brengt Jacobson direct verslag uit over de conferentie. “We hebben een kleine kans om de wereld te verbeteren met Objectory AB”, meldt hij opgewekt aan zijn medewerkers; gerechtvaardigd gezien de feedback van de OOPSLA. “Een van mijn medewerkers vroeg me bij deze gelegenheid hoe groot ik die kans inschatte. Ik hoefde er niet lang over na te denken. Ik schatte onze kans om de wereld te verbeteren in op drie procent”, vertelt hij met glinsterende ogen.

Het OMG wilde vooral een modelleertaal, die onafhankelijk was van het gebruikte systeemontwikkelproces

face introduceerde ten behoeve van de embedded software engineering. Niet lang nadat Jacobson in aanraking komt met objectoriëntatie start hij zijn eigen bedrijf, Objectory AB. Al snel ziet de grijzende Zweed het belang in van het beschikken over een systeemontwikkelproces voor projecten waarin objectoriëntatie een belangrijke rol speelt, zo rond 1987. “Bij Objectory AB ontwikkelden we zo het Objectory Process. Dit was

**“USE CASES CAME TO ME IN A FLASH”** Aan het einde van de jaren tachtig ontstonden er naast Objectory diverse andere nieuwe methoden en technieken voor objectgeoriënteerde software engineering. Bekend zijn de methoden van Shlaer/Mellor, van Coad/Yourdon, maar ook de Booch Method en OMT van James Rumbaugh. “Ieder van deze methoden kende zijn eigen concepten, notaties en processen”, zegt Jacobson. Alhoewel ieder van de nieuwe methoden dezelfde objectgeoriënteerde software engineering beschreef ontstond er binnen de kortste keren een explosie aan concepten en notaties, niet zelden overlappend. Dit leidde in de objectgeoriënteerde gemeenschap tot het inzicht dat een algemeen geldende verzameling concepten en notaties geen overbodige luxe was, wilde het vakgebied verder rijpen. “Het Objectory Process was een opvallende eend in de bijt. Bij de meeste methoden stond het vinden van de belangrijke klassen centraal. Het Objectory Process stelde het bedrijfsproces echter als uitgangspunt.”

“Om het bedrijfsproces te beschrijven introduceren we het begrip use case”, gaat Jacobson verder. Als dagelijkse gebruiker van use cases vraag ik hem naar het ontstaan van de modelleertechniek. “It came to me in a flash”, glimlacht de methodoloog. “De use case komt min of meer voort uit mijn periode bij Ericsson. Daar kende men het begrip traffic cases. Dit werd gebruikt voor de analyse van de werking van onder meer telefooncentrales. Later realiseerde ik me dat ditzelfde begrip direct vertaalbaar was naar objectgeoriënteerde software engineering.”

De rest is geschiedenis. Enkele jaren later kent het bedrijf Objectory AB inmiddels vestigingen in Duitsland, Zweden en de Verenigde Staten. Het Objectory Process is dan gevorderd tot versie 4.0. Op initiatief van Grady Booch en de inmiddels eveneens voor Rational werkende James Rumbaugh wordt Objectory AB overgenomen door Rational. De ideeën en modelleertechnieken die Ivar Jacobson in Objectory heeft verwerkt, zoals use cases, sequence diagrammen en collaboration diagrammen, vinden snel hun weg bij de leverancier van gereedschappen. Uit de samenvoeging van het Objectory Process met OMT van James Rumbaugh en de Booch Methode evolueert de eerste versie van de Unified Method.

**UNIFIED MODELING LANGUAGE** De Object Management Group (OMG) is een internationale organisatie die standaarden beheert, ontwikkelt en promoot op het gebied van objectoriëntatie. De Object Management Group richt zich hierbij op doelen als herbruikbaarheid, overdraagbaarheid en interoperabiliteit van software. De organisatie telt honderden leden, waaronder software-ontwikkelaars, leveranciers van



Sander Hoogendoorn (r) in gesprek met Ivar Jacobson

ontwerp- en ontwikkelgereedschappen en eindgebruikers van de standaarden. Net na de overname van Objectory AB door Rational vaardigt de Object Management Group een request for proposal uit voor een standaard voor objectgeoriënteerde modellering. Uiteraard leveren de three amigos bij Rational belangrijke ideeën voor de voorstellen die hiervoor worden ingediend bij de OMG. Omdat de OMG vooral geïnteresseerd is in een modelleertaal, onafhankelijk van het gebruikte systeemontwikkelproces, splitsen Jacobson, Rumbaugh en Booch, het Objectory proces af van de gebruikte modelleertechnieken. Zo ontstaat het Rational Unified Process, kortweg RUP. De modelleer-

Jacobson: “I feel Rational is my company.  
Even if it isn’t”

technieken van Ivar Jacobson worden integraal opgenomen in de modelleertaal, die in november 1997 de Unified Modeling Language wordt gedoopt.

**AGILE DEVELOPMENT** Het gesprek verschuift van onderwerp en intensiteit wanneer ik Jacobson vertel dat ik zijn ideeën en modelleertechnieken dagelijks gebruik. Maar meestal wel in een eigen systeemontwikkelmethode, die behoort tot de nieuwe generatie agile systeemontwikkelmethoden [Ordina SMART, red.]. Benieuwd naar zijn mening vraag ik hem hoe hij denkt over deze nieuwe generatie methoden en extreme programming in het bijzonder. Het gezicht van de goeroe verstrakt iets, alhoewel het vriendelijk blijft. “Laat ik hier heel duidelijk in zijn”, legt hij uit. “De hele bewe-

ging naar agile development is waarschijnlijk het beste dat ons [Rational, red.] op dit moment kan overkomen. Het helpt ons na te denken over agility. Het is niet dat we niet hebben nagedacht over agility bij het ontwikkelen van het Rational Unified Process. Integendeel, het begrip agility speelt altijd door mijn gedachten. Maar RUP is vooral ontwikkeld voor het bereiken van kwaliteit in systeemontwikkelprojecten, door precies te vast te leggen hoe je te werk gaat. En daarbij lag de nadruk niet altijd op agility. We zien nu dat het nodig is om meer agile te worden.”

“Kijk”, gaat de Zweed verder, “RUP is ontegenzeggelijk de grootste verzameling consistent beschreven kennis ter wereld op het gebied van software engineering. Er zijn ongeveer vijftientig tot dertig boeken verschenen waarin de verschillende aspecten van RUP worden toegelicht. En dat allemaal in dezelfde terminologie en in dezelfde vocabulaire. Al deze boeken zijn nauw geïntegreerd met elkaar. Zelfs de manier waarop we Rational Unified Process hebben beschreven is uitgegroeid tot een standaard voor het beschrijven van systeemontwikkelprocessen. RUP is een enorme en fantastische database van kennis.”

**PROCESONDERSTEUNING** Jacobson denkt even na en vervolgt zijn betoog. “Natuurlijk leest niemand alle boeken die er over het onderwerp geschreven zijn. Maar

dat hoeft ook niet. Je weet dat het werkt. Een proces moet overigens wel ondersteund worden door gereedschappen, bijvoorbeeld voor het beheren van requirements of het modelleren van functionaliteit. Als er geen gereedschappen zijn om een proces te ondersteunen, kan het nooit succesvol worden geïmplementeerd in een organisatie.” Jacobson laat zich in zijn betoog duidelijk inspireren door het feit dat er naast Rational Unified Process ook andere processen worden ingezet in projecten. “Toen we bij Objectory AB ons proces ontwikkelden, hoorde het inzetten van gereedschappen niet tot de uitgangspunten. We waren primair gericht op het beschrijven van het proces. Maar ons proces moest worden ondersteund door gereedschappen. Rational bood in 1995 primair dergelijke gereedschappen. Dus toen Rational mijn bedrijf overnam, kwamen de gereedschappen beschikbaar om het proces te ondersteunen.”

**OVERNAME** Ik onderbreek het betoog van de methodologist en vraag hem hoe hij het ervaren heeft om overgenomen te worden door Rational. “Ik haatte het”, lacht de Zweed breeduit, “maar ik raakte al snel overtuigd van het nut van de overname. Het was het beste dat kon gebeuren met Objectory AB. De beide ondernemingen hadden op dat moment heel verschillende uitgangspunten. Vanuit Objectory AB hadden we

---

**ADV. EXCELON**

al geïnventariseerd dat er voor succesvolle software engineering drie aspecten van groot belang waren. Je dient te beschikken over een goed systeemontwikkelproces, en daarnaast een verzameling ondersteunende gereedschappen. Bovendien heeft een organisatie behoefte aan services om zowel het proces als de gereedschappen te ondersteunen. Anders zul je niet slagen.

“Vergis je niet in de impact die de overname van Objectory ook op Rational heeft gehad. Rational was in 1995 een tools company pur sang. Bij Objectory AB hadden we zo’n veertig consultants, verspreid over Zweden, Duitsland en de Verenigde Staten, die vooral actief waren in het implementeren van ons proces. Nog voor de overname hadden we geconstateerd dat we de verdere ontwikkeling van het Objectory Process moesten stopzetten als we het niet konden ondersteunen met gereedschappen”, herhaalt Jacobson nog maar eens. “De overname kwam precies op het goede moment. En kijk maar waar Rational heden ten dage staat. We bieden een proces, de gereedschappen om het proces te ondersteunen en bieden ook alle ondersteunende services. De ontwikkeling van Rational Unified Process werkt als katalysator voor de ontwikkeling van de diverse gereedschappen. Maar het werkt ook vice versa. De continue verbetering van de gereedschappen is extreem stimulerend op de ontwikkeling van het proces.”

De goeroe staart naar buiten en lijkt de jaren sinds de overname te overpeinzen. “I feel Rational is my company,” concludeert hij tenslotte. “Even if it isn’t.” Jacobson lacht. De public relations dame aan zijn rechterhand lacht met hem mee, zij het met licht andere intonatie.

**CONCURRENTIE** Tijd om Jacobson bij de les te halen. Hij heeft zich nog niet uitgesproken over de nieuwe generatie agile systeemontwikkelmethoden en over extreme programming in het bijzonder. Toch zal de concurrentie van deze nieuwe generatie voor Rational zeer voelbaar zijn. “Rational Unified Process is een goed proces”, zegt de Zweed. “Het is configureerbaar naar de behoeften van een organisatie. Het proces kan worden geschaald naar gelang deze behoeften. Gebruik daarbij de kennisbank van RUP zoveel als je nodig vindt. Ook met RUP kun je agile zijn. Voor de implementatie van een agile proces heb je trouwens nog altijd ondersteunende gereedschappen nodig. Daar is niets nieuws aan, toch? Alhoewel het is bedoeld om betere kwaliteit software op te leveren, vertelt extreme programming je niet hoe je precies moet ontwerpen en programmeren. De kwaliteit van code verbetert alleen als de gereedschappen die je gebruikt beter en beter worden, terwijl je daarbij meer en meer kennis over het proces verzamelt. Extreme programming doen zonder

te beschikken over gereedschappen blijft een kaal proces.”

**VOORTBORDUREN OP RUP** Jacobson grijpt weer even terug op zijn lange loopbaan. “Al in 1989 kenden we voor het Objectory Process verschillende varianten, al naar gelang de behoefte van de klant. We hadden Objectory Light, Objectory Premium en een variant voor het beheren van requirements in projecten. Er was nog een vierde variant waarvan de strekking me nu

Fowler: “Since RUP can be anything, it often ends up being nothing”

even niet te binnen schiet. We noemden deze varianten development cases. Ons bedrijf Objectory AB groeide echter snel met als gevolg dat de processen moesten worden herontworpen. Het grote probleem hiermee was dat we indertijd niet beschikten over gereedschappen om dit te doen. Toen we werden overgenomen door Rational besloten we om de ontwikkelaars zelf te laten kiezen. Er moest nog gereedschap worden ontwikkeld om het proces aan te passen aan ieders individuele behoefte. Rational heeft dergelijke gereedschappen nu. Nu is het dus wel mogelijk om development cases te maken.”

Nu wordt Jacobson serieus. “Als ik je een advies mag geven”, zegt de goeroe op onderwijzende toon, “don’t do your own process. Gebruik Rational Unified Process



Ivar Jacobson: ‘Als ik je een advies mag geven...don’t do your own process’

en borduur hierop voort. Dat is de manier om succesvol te zijn in software engineering. Het zelfstandig ontwikkelen van een proces is nagenoeg onmogelijk. Het kost enorm veel energie en tijd. Een organisatie moet serieus overwegen of het de moeite loont een eigen proces te ontwikkelen en te onderhouden. Misschien verlaat de

## ‘Een organisatie moet serieus overwegen of het de moeite loont een eigen proces te ontwikkelen’

auteur van het proces de organisatie wel. Misschien krijgt hij een betere baan aangeboden van Rational.” Jacobson glimlacht. “Wees realistisch”, vervolgt hij. “Keep up your ego. Haal alles uit RUP waar je maar bij kunt. Gebruik het als uitgangspunt en borduur erop voort. Wellicht ondersteund door nieuwe technologieën als intelligent agents.”

**VERVOLG** Een goed onderwerp voor het tweede deel van ons gesprek. Jacobson is een commercieel denker. De verkoper van zijn eigen ideeën. Het implementeren van het Rational Unified Process is echter verre van een-

voudig. Het ontwikkelen van een door Jacobson voorgestelde development case vergt kennis en ervaring op velerlei gebied. Organisaties die falen in het inrichten van een eigen development case, kiezen er vervolgens maar al te vaak voor om het hele proces te implementeren. Voorwaar geen sinecure. “RUP is vooral een heleboel”, vertelde een gebruiker onlangs. Martin Fowler, mede-opsteller van het agile manifesto, gaat in een recent artikel nog verder. “Since RUP can be anything, it often ends up being nothing”, verzucht Fowler.

*Het tweede deel van dit gesprek volgt in het volgende nummer van Software Release Magazine. Onderwerpen van gesprek zijn dan onder meer het inzetten van intelligent agents en het vertalen van ontwerp naar code.*

*Sander Hoogendoorn  
Partner bij Ordina en chief architect van de agile systeemontwikkelmethode Ordina SMART. sander.hoogendoorn@ordina.nl.*

---

**ADV. DELTA SOFTWARE**