Ivar Jacobson vervolgt nog eenmaal zijn betoog over agile development en Rational Unified Process (zie Software Release nr. 4, juni 2002). "The Rational Unified Process is agile enough", stipuleert de Zweed nogmaals. De Engelse account manager van Rational die het gesprek tot nu toe vanaf de zijlijn heeft gevolgd neemt op het goede moment het heft in handen. Ze stelt voor het gesprek te voort te zetten aan de andere kant van de Brasserie van hotel Karel V in Utrecht, voor de open haard. Jacobson stemt toe en we verplaatsen naar de warme en comfortabele fauteuils voor het brandend haardvuur.

interview

Ivar Jacobson: "I count my ideas"

Over use cases, intelligent agents en executeerbare modellen

Snel verlaten we het onderwerp agile development. Hier gaan we het niet eens worden. Om het gesprek te wenden vraag ik Ivar Jacobson hoe hij in zijn lange loopbaan zo gemotiveerd blijft. De Zweedse grondlegger van UML lacht breeduit. "Natuurlijk verdien ik bij Rational niet slecht", verklapt hij, "maar dat is nooit mijn drijfveer geweest. Ik haal mijn motivatie uit mijn ideeën. I count my ideas."

Ik vraag Jacobson welk van zijn ideeën hij het meest belangwekkend vindt. Ik doe hem alvast de suggestie use cases aan de hand. Heel even denkt Jacobson na. "Ik denk dat ik use cases, en de aanverwante modelleertechnieken als sequence diagrams en activity diagrams op de tweede plaats zou zetten. Het zijn hele nuttige technieken, die in de wereld van software engineering grote verspreiding en acceptatie kennen", gaat hij verder, "maar ik vind het niet mijn belangrijkste idee. Ik denk namelijk dat ik het concept component en interface op de eerste plaats zou zetten. Al in 1976 schreef ik, toen nog in dienst van Ericsson, een eerste artikel waarin ik de concepten component en interface benoemde. Nog ver voor ik in aanraking kwam met objectgeoriënteerde software engineering. In de telecom werd al langer met componenten gewerkt. De concepten component en interface hebben nog een veel grotere impact op het vakgebied software engineering dan use cases."

PRATEN MET INTELLIGENT AGENTS "En op de derde plaats", zegt Jacobson glunderend, "plaats ik een relatief nieuw idee. Dit heeft te maken met het inzetten van intelligent agents in het toepassen van een systeemontwikkelmethode zoals Rational Unified Process." De Zweed schakelt naar een hogere versnelling. "Twintig jaar geleden schreef ik hier al over. Dit is mijn droom. Als je weet hoe je moet modelleren, hoe het werk in projecten geschiedt en hoe je deliverables er uit zien, dan kun je je laten ondersteunen door een expertsysteem. Twintig jaar geleden was dit onmogelijk. Nu zijn er expertsystemen die dat voor je zouden kunnen doen. Alhoewel ik met dit idee constant heb rondgelopen, begon ik er enkele jaren geleden weer aan te denken."

De luxe fauteuils in de Brasserie doen het goed als praatstoel. Ivar Jacobson wordt steeds enthousiaster en buigt iets naar mij toe om zijn verhaal nog beter te articuleren. "We hebben de ideeën bediscussieerd binnen Rational. Het juiste tijdstip is nu aangebroken om dit idee in de praktijk te brengen. Alle voorwaarden zijn inmiddels vervuld. We hebben een modelleertaal en

> Jacobson: "Je maakt nu eenmaal het beste ontwerp als je denkt als een tester"

een systeemontwikkelproces. Zo weet je hoe een applicatie eruit ziet en hoe je systeemontwikkeling doet. Tijd dus om je te laten ondersteunen door experts."

Microactiviteiten Jacobson houdt even in, als om duidelijk te maken dat wat hij nu gaat zeggen van groot belang is voor het vakgebied en misschien nog wel breder. "Voor ieder artifact dat je oplevert, voor iedere activiteit die je uitvoert en voor iedere worker - zeg maar rol - in Rational Unified Process komt er een intelligent agent die een expert is op het gebied van die ene artifact, over de rol die we hebben in het project, over wat de andere projectmedewerkers doen. Niet alleen de basale zaken in een project, maar inclusief alle microactiviteiten die er zijn in projecten."

> "Geen ontwikkelgereedschap kan je de ondersteuning bieden die een intelligent agent biedt"

De Zweed maakt even een zijsprong om te verklaren wat microactiviteiten zijn. Iedere handeling die een projectmedewerker doet tijdens een project is zo'n microactiviteit. Ieder patroon van modelleren, van het bedienen van de modelleergereedschappen tot aan het toepassen van design pattern toe. "Misschien zijn er al wel zo'n vijfhonderd tot zevenhonderd van die microactiviteiten alleen al in het modelleren", gaat Jacobson verder. "Vaak weet je niet eens dat je ze uitvoert. We moeten dus mensen in projecten bestuderen om deze microactiviteiten te vinden. Neem mensen daarvoor bijvoorbeeld op video op. Zo kun je precies observeren wat hij of zij doet. Identificeer vervolgens de microactiviteiten."

Sander Hoogendoorn (r) in gesprek met Ivar Jacobson

Creatief werk "Wist je dat zo'n beetje tachtig procent van al het werk in systeemontwikkelprojecten als creatief werk wordt beschouwd?", vraagt Ivar Jacobson mij na even peinzend in het haardvuur gekeken worden. Het is warm in de Brasserie. "Maar heel veel van dat werk kan worden gedaan door intelligent agents. Ze kunnen je helpen bij het opstellen van de modellen door met jouw te redeneren over het hoe en waarom van het modelleren. Agents kunnen je bijvoorbeeld steeds opties en alternatieven aanreiken."

"De agent observeert jouw microactiviteiten en kan je bijvoorbeeld tijdens het modelleren vragen of je een bepaalde pattern probeert te gebruiken, en je daarbij vervolgens assisteren. De intelligent agents weet wat jij aan het doen bent. Agents kunnen patterns precies op tijd toepassen, en bovendien geautomatiseerd. Ze ondersteunen je bijvoorbeeld bij het opstellen van use cases en de bijbehorende beschrijvingen."

Schijnbaar kijk ik wat ongelovig, want Jacobson meldt dat dergelijke werkwijzen dichtbij zijn. "This is not a fantasy", zegt hij. "This is really working now. Er wordt op dit moment aan gewerkt. Rational doet overigens niet al dit werk zelf, maar werkt hiertoe samen met verschillende partners. We proberen ervoor te zorgen dat je niet zelf dingen doet die al beschreven zijn."

Alhoewel ik graag geloof dat intelligent agents het complexe werk van het modelleren van software kunnen verlichten, bedenk ik me dat de manier waarop dit gebeurt de ontwerpers met handen en voeten aan de zienswijze van Rational bindt en inderdaad minder ruimte laat aan creativiteit en andere manieren van modelleren. "Gebruik je kennis en ervaring om samen met agents je werk bovenop RUP te doen. De intelligent agents doen precies hetzelfde", gaat Jacobson verder, en brengt daarmee mijn vrees onder woorden. "Geen ontwikkelgereedschap kan je de ondersteuning bieden die zo'n intelligent agent biedt", zegt hij tenslotte.

Executable code Naast het haast science fiction-achtige idee van de agents voor het Rational Unified Process heeft de Zweed nog een toekomstdroom. "Making UML an executable code", noemt Jacobson het. Tussen het maken van een ontwerp in UML en het ontwikkelen van code zit een gat. Dit wordt wel het semantisch gat genoemd. "UML gaat niet verder dan de signatures van de methoden van klassen. Dat is het punt waar re-engineering en reverse engineering begint en eindigt. Er moet nog altijd veel code worden geschreven." Liever ziet Jacobson dat code wordt ontwikkeld in de UML ontwerpomgeving.

Als ik hem enigszins vragend aankijk legt hij het idee uit. "Bij Ericsson ontwikkelden we dertig jaar geleden al componenten voor de telecomindustrie. Deze modelleer-

den we met activity diagrams, collaboration diagam en state charts. In die tijd was er een standaard modelleertaal voor de telecom, die SDL heet. Veel van de ideeën daarin komen van Ericsson. SDL is tegelijk een programmeertaal." Even stopt Jacobson om te verifiëren of de koffie inderdaad op is. "Het executeerbaar maken van modellen is dus een oud idee. Waar zou dit niet ook met UML kunnen? In de telecom zal UML binnen een paar jaar SDL vervangen. UML is tenslotte superieur aan SDL."

"Als je nog tien procent extra aan UML toevoegt om het gat tussen ontwerp en code te dichten, dan kun je code genereren vanuit het model in architecturen als Java/J2EE of .NET. Rational gaat dit echter niet zelf doen. Het wachten is op een eerste business case. Misschien duurt het nog wel tien jaar voordat we zover zijn, maar er is geen fundamentele reden waarom het niet zou kunnen."

Ivar Jacobson: 'Testing belongs where you are'

DENKEN ALS EEN TESTER De account manager van Rational is terug en wijst op haar horloge. Jacobson neemt nog de tijd voor een laatste stokpaardje. Testen. "Idealiter verifieer je alles dat je oplevert tijdens een project. Er is een veelheid aan mechanismen voor het testen van bijvoorbeeld de requirements en het ontwerp van functionaliteit. Testing belongs to where you are. Het uitvoeren van tests is nauw verbonden met datgene dat je test. Iedere ontwerper en ontwikkelaar zou ook tester moeten zijn."

"In veel organisaties werkt dit niet zo. Er zijn mensen die iets doen, en er zijn mensen die opruimen. Maar mensen die iets doen zouden zelf verantwoordelijk moeten zijn voor het opruimen van wat ze achterlaten. Clean up after yourself. Je voert een taak uit, ruimt en verifieert of alles is opgeruimd. Als iemand anders binnenkomt, weet hij waarschijnlijk niet precies wat jij net gedaan hebt."

"Je maakt nu eenmaal het beste ontwerp als je denkt als een tester. Dus als je verantwoordelijk bent voor het opstellen van use cases, dan ben je ook verantwoordelijk voor het testen van de use cases." Jacobson houdt zich even in. "Maar ik ben hier wellicht wel wat extreem in. Ik ben me er terdege van bewust dat een dergelijke werkwijze voor veel mensen een flinke verandering betekent. Misschien moet je voor het testen van use cases wel sequence en collaboration diagrams ontwikkelen."

Het is niet voor niets dat in agile systeemontwikkelmethoden als XP en Smart gebruikt wordt gemaakt deze veranderende verantwoordelijkheden. Beide methoden maken gebruik van unit testing. Hierbij bedenken de ontwikkelaars eerst test classes. Aan de hand daarvan realiseren ze vervolgens de functionaliteit. In Smart worden bovendien de use cases vroegtijdig getest aan de hand van activity diagrams die worden opgesteld door samenwerkende gebruikers en testers. Hier is het adagium van Jacobson vice versa toegepast. Testers zijn ook ontwerper.

DE TIJDGEEST Ivar Jacobson is een markante persoon met een zeer belangwekkende loopbaan. Een loopbaan waar de Zweed wel erg prat op gaat. Jacobson als bedenker van use cases. Jacobson de uitvinder van component en interface. Jacobson als grondlegger van de Unified Modeling Language. En vooral Jacobson de vader en voornaamste promotor van Rational Unified Process, als allesomvattende en door intelligent agents te ondersteunen kennisbank over software engineering. Zelfs als het proces niet altijd even gemakkelijk implementeerbaar is en de tijdgeest van pragmatische en agile systeemontwikkeling tegen zich heeft. Het tijdperk van de gefaseerde en gedetailleerde methodieken is wellicht voorbij. Het allesomvattende daarvan werkt nogal eens verstikkend op diverse organisaties.

> Het tijdperk van gedetailleerde methodieken is wellicht voorbij: het allesomvattende daarvan werkt vaak verstikkend

Ik geef Ivar Jacobson een hand en loop op het parkeerterrein van hotel Karel V de heldere, zonnige, maar koude Utrechtse middag in.

Dit is het tweede en laatste deel van het interview met Ivar Jacobson. Het eerste deel van dit gesprek is gepubliceerd in Software Release Magazine nr. 4, juni 2002.

Sander Hoogendoorn, patner Ordina
(e-mail: sander.hoogendoorn@ordina.nl)