

‘Hergebruik is dertig jaar lang heel erg onsuccesvol geweest’

Producten baseren op een los onderhouden platform biedt veel betere mogelijkheden tot softwarehergebruik dan een platform baseren op de producten, betoogt Jan Bosch. Hij zag deze productlijnaanpak begin jaren negentig opkomen en is er sindsdien actief in. Grote Europese bedrijven als Philips Healthcare en Nokia hebben er de afgelopen tien jaar hun voordeel mee gedaan. Ondertussen werkt Bosch in Silicon Valley, waar hij de volgende grotere verandering ziet aankomen. De manier waarop we software bouwen, zal ingrijpend veranderen. Dat begint op het web, maar zal uiteindelijk zijn weerslag hebben op embedded systemen.

Pieter Edelman

Hergebruik. Bij het grootste deel van de bevolking zal het woord vooral connecties oproepen met de glasbak en het ophaalrooster voor oud papier. De softwarebetekenis van het woord werd echter al in 1968 besproken op de Navo-conferentie voor software-engineering in het Duitse Garmisch, vier jaar voordat de Club van Rome de aftrap gaf voor de milieubeweging. Toch is softwarehergebruik vier decennia later bij veel bedrijven nog steeds een worsteling, constateert Jan Bosch. Hij is een van de drijvende krachten achter een hergebruikbenadering die wel goed blijkt te werken: *software product lines* (SPLs). Kort gezegd gaat het hierbij om een op zichzelf staand platform dat dient als basis voor verschillende producten.

‘Nokia ontwikkelt op dit moment denk ik zo’n vijftig mobiele telefoons per jaar. Vijftien jaar geleden waren dat er vier of vijf. Dat het er nu tien keer zo veel zijn, komt doordat ieder nieuw telefoontype wordt ontwikkeld op basis van een standaard architectuur en een standaard set van componenten, die los van de productontwikkeling wordt onderhouden’, legt Bosch het basisidee uit. ‘Je ziet dat hergebruik van software binnen een bedrijf eigenlijk dertig jaar lang heel erg onsuccesvol is geweest. Wat hebben we gehad? Kleine componenten, objecten, *object-oriented frameworks*, *component-based software engineering*. Geen van die benaderingen heeft echt de verwachte voordelen gebracht. De reden van het succes bij SPLs

is dat ze niet alleen naar technologie kijken, maar ook naar de bedrijfsstrategie die ermee wordt geïmplementeerd, de architectuur en de technologiekeuzes. De ontwikkelprocessen binnen de organisatie worden expliciet meegenomen, we kijken zelfs naar de organisatie van de R&D. Pas toen al die verschillende aspecten werden meegenomen, is het hergebruik binnen organisaties succesvol geworden. Dat is ongeveer midden jaren negentig op gang gekomen, met de grootste successen in laatste tien jaar.’

Toen de aanpak begin jaren negentig opkwam, werkte Bosch aan het Zweedse Blekinge-instituut als hoogleraar software-architectuur. Het idee om niet alleen naar de technische kant te kijken, intrigeerde hem en hij omhelsde SPLs met beide armen. Via diverse bedrijven – waaronder Nokia – is hij nu in Silicon Valley terechtgekomen bij Intuit, dat financiële software ontwikkelt. Daar vormen productlijnen nog steeds het belangrijkste deel van zijn werkzaamheden, niet alleen in zijn dagelijkse baan maar ook

die allemaal op basis van dezelfde architectuur zijn ontwikkeld. Bij Bosch in Duitsland zijn de *engine control units* allemaal gebouwd op basis van een product line, en zo zie je dat heel veel grote Europese bedrijven heel erg succesvol zijn geweest juist door het adopteren van een SPL-aanpak.’

Cash cow

Het succes van softwareproductlijnen is tot nog toe vooral een feestje van de grote Europese bedrijven. Maar het is tijd dat daar verandering in komt, meent Bosch. Ook middelgrote bedrijven kunnen grote voordelen halen uit SPLs, mits ze om de veelvoorkomende valkuilen weten heen te sturen. Om die te begrijpen, is het nodig eerst even een stapje terug te doen en te kijken hoe de ontwikkeling van een productlijn in zijn werk gaat. ‘Typisch zie je binnen bedrijven een evolutieproces waarbij ze een succesvol product ontwikkelen, daarna een tweede, daarna een derde, en dan beginnen ze te zien dat er een aanzienlijke overlap is. Terwijl ze de

‘Ieder nieuw platform werd aangekondigd als de moeder van alle platforms’

als consultant. De afgelopen jaren heeft hij verschillende bedrijven bijgestaan bij het implementeren van productlijnen. ‘Bij Philips Healthcare kun je bijvoorbeeld heel veel verschillende medische apparaten bestellen

gemeenschappelijke functionaliteit voor al die producten steeds weer opnieuw ontwikkelen en voor al die producten apart moeten onderhouden. Dan gaan ze vaak op zoek naar een vorm van gemeenschappelijk



Jan Bosch

gebruik van de softwarebasis. Daarmee kunnen ze veel eenvoudiger producten ontwikkelen of bestaande producten sneller uitbreiden met functionaliteit.'

'Dan kom je bij de eerste vraag: welke businessdoelen moeten ermee worden bereikt? Er is een discussie nodig om dat duidelijk geformuleerd te krijgen. Als er dan een strategiekeuze is gemaakt, begint de exploratiefase waarin wordt geëxperimenteerd met het concept van een nieuw platform. Als de eerste prototypes en experimenten succesvol zijn, wordt er een beslissing genomen en gaat de ontwikkeling over van de exploratie- naar de initiatiefase. Dat gebeurt relatief hoog in de organisatie, want het in-

vesteringsniveau moet flink omhoog terwijl er nog geen omzet wordt gegenereerd.'

'Op een gegeven moment verschuiven de bronnen van het huidige naar het nieuwe platform. Dit is vaak de meeste gevaarlijke fase voor een bedrijf, want juist in de periode dat je wel investeert maar geen inkomsten hebt, ben je kwetsbaar. Aan het eind van deze fase komen dan de eerste producten op de markt en begint de uitbouwfase. De investering gaat nog verder omhoog, maar er komen ook inkomsten. Die eerste producten zitten vaak aan de rand van de segmenten waarin het bedrijf actief is omdat ze hun bestaande corebusiness niet in gevaar willen brengen met een nieuw platforminiti-

atief. Maar als de eerste producten hun succes in de markt hebben bewezen, zie je dat de business steeds meer bereid is om producten te ontwikkelen die naar de kern van hun markt gaan.'

'En dan krijg je de *institutionalized phase*: de investeringen zijn gedaan, de inkomsten groeien, de productlijn begint steeds meer bestaande marktsegmenten en ook nieuwe segmenten te ondersteunen. De oude softwareartefacten blijven dan vaak nog wel bestaan als *cash cow*, maar de investeringen erin zijn vele malen lager.' Als voorbeeld noemt Bosch Nokia, dat nog steeds de oudere S30- en S40-platforms inzet naast het modernere S60 en Linux-gebaseerde MeeGo. 'Iedere keer als er een nieuw platform kwam, werd dat aangekondigd als de moeder van alle plat-

forms, die alles zou gaan vervangen. Maar in de praktijk zag je dat de oude platforms niet verdwenen. Series 30 wordt nog steeds gebruikt voor de low-end, met name de opkomende markten.'

Erosie

In de lange aanloopfase schuilt een van de problemen voor de relatief kleine bedrijven. 'Maar er is een manier om die initiële investering terug te brengen, namelijk door in het begin wel voor de hele reeks van beoogde toepassingen te ontwerpen, maar alleen de functionaliteit te bouwen voor producten die je als eerste op de markt wilt brengen. Als je dat in een goede balans doet, is de

voorafgaande investering een stuk lager dan wanneer je eerst alles moet gaan bouwen.'

Kortom, probeer niet alles in één keer goed te doen, probeert Bosch maar te zeggen. 'Met name de mensen die gefrustreerd zijn over beperkingen van oude software zijn zo bezorgd over het afsnijden van kantjes dat ze het nu in één keer goed willen doen. Ze beginnen vanuit de fundamentgedachte, willen eerst de basiscomponent hebben waar ze een laag bovenop bouwen, en daar weer een laag bovenop en nog een laag... Dan is de investering vaak onrealistisch hoog en overleven de initiatieven het vaak niet.'

Een andere valkuil is de verleiding om geen geld uit te geven aan het onderhouden van de productlijn. 'Bedrijven als Philips hebben als vuistregel dat er zo'n tien procent van het totale budget moet worden gebruikt voor onderhoud, om het erosieniveau ongeveer constant te houden. In de institutionalized phase is het nieuwe platform echter zo centraal voor de groei en omzet van het bedrijf dat vrijwel alle investeringen naar nieuwe functionaliteit gaan. In de praktijk is het vaak moeilijk om de businesskant te overtuigen dat langetermijninvesteringen noodzakelijk zijn voor onderhoud en het aanpassen aan toekomstige eisen.' Dat er vaak enige frictie is tussen de R&D- en businessafdelingen maakt het er wat dat betreft niet makkelijker op. 'Dat vind ik buitengewoon ongezond. Juist bij product lines wordt het succes of het falen bepaald door de mate waarin R&D en business met elkaar in overeenstemming zijn. Dus moet je actief omgaan met dat gebrek aan vertrouwen.'

Mits goed onderhouden, kan een softwareproductlijn de basis vormen voor een brede reeks producten, maar uiteindelijk zal die ook niet meer voldoen. 'Zo ver als je de toekomst kunt voorspellen, wordt er heel expliciet nagedacht over wat de richting is van een productlijn. Wat zijn de producten die we waarschijnlijk willen bouwen? Waar liggen de technische limieten? Hoe flexibel moet het zijn? De levensduur van een productlijn is daardoor langer dan van een individueel product, maar niet oneindig. Iedere vijf tot tien jaar is er zo'n grote verschuiving in de beperkingen van de technologie ontstaan dat de architecturale beslissingen voor het oorspronkelijke platform te beperkend zijn voor de nieuwe situatie.'

Kathedraal

Bosch begon zijn carrière academisch, eerst in Zweden en vanaf 2000 aan de Groningse universiteit. Daar smeedde hij de banden met het bedrijfsleven. Hij werkte onder meer nauw samen met Bosch, Ericsson, Nokia, Philips, Siemens en Thales. In 2004 volgde de definitieve overstap naar het bedrijfsleven met een aanstelling bij Nokia, als hoofd van een van de Finse researchlabs, het Software and Application Technologies Laboratory,

dat destijds zo'n tweehonderd werknemers telde. Twee jaar geleden deed Intuit een bod. Dat was een nieuwe SPL aan het opstarten en kon wel wat externe expertise gebruiken.

Vanuit die positie als softwareman in Silicon Valley ziet Bosch in rap tempo een ingrijpende ontwikkeling ontvouwen als vervolgstap op productlijnen: software-ecosystemen. 'Die zie ik in Silicon Valley heel hard opkomen en ik verwacht dat zij zich breder en verder zullen verspreiden.' Dat zal de manier van software ontwikkelen

ders te gaan doen. Amazon gebruikte de *two pizza rule*: geen team mag groter zijn dan wat je met twee pizza's kunt voeden als ze heel erge trek hebben. Dus vaak teams van maximaal zes tot zeven mensen.'

Bosch denkt dat hier ook zeker een invloed te merken is van de steeds professioneler wordende opensourcebeweging. Gedistribueerd ontwikkelen is daar immers de regel. 'Bedrijven zijn verbaasd over de efficiëntie waarmee de opensourcegemeenschap dingen kan ontwikkelen. Al dat soort

'Voor traditionele benaderingen als CMMI zie ik gewoon de end of the road'

ingrijpend veranderen en de rol van de architectuur op zijn kop zetten, denkt Bosch. 'We gaan naar een veel meer compositiegebaseerde manier van software ontwikkelen toe in plaats van de traditionele, integratiegerichte benadering, met grote teams die manueel al de verschillende componenten integreren, met soms honderden of duizenden mensen aan het werk aan één systeem, en waar roadmapping en requirementsmanagement buitengewoon complex zijn en veel moeite kosten. Ik geloof echt dat we ons op een *inflection point* in software-engineering bevinden. Voor traditionele procesbenaderingen als CMMI zie ik gewoon de *end of the road*.'

Met software-ecosystemen doelt Bosch op de toenemende trend om programmeerinterfaces en ontwikkelkits te publiceren rondom een platform voor derde partijen, zodat zij er applicaties voor kunnen ontwikkelen. 'Je ziet in heel veel industrieën, zowel op het web als bij embedded, dat klanten steeds meer eisen gaan stellen aan individuele configureerbaarheid van producten. Ze willen niet alleen maar instellingen kunnen veranderen, maar een stap verder kunnen gaan en een eigen oplossing kunnen samenstellen uit verschillende *bits en pieces*. Neem Facebook, daar heb je zo'n tienduizend applicaties waarmee je je eigen homepage kunt uitbreiden met verschillende applicaties. Dat zie je ook rond de Iphone. Steeds meer bedrijven bouwen een basisplatform en proberen daaromheen een ecosysteem van externe ontwikkelaars te vormen, met een businessmodel waarbij die een redelijke hoeveelheid geld kunnen verdienen.'

Als zo'n basissysteem er eenmaal ligt voor externe ontwikkelaars, dan ligt het natuurlijk ook voor de hand om de *interne* ontwikkeling daaromheen te organiseren. En dat betekent kleine onafhankelijke groepjes. 'Hier in Silicon Valley zie je dat al veel. Google heeft bijvoorbeeld heel lang het principe van drie keer drie gebruikt: teams van drie personen die drie maanden werkten aan één product of component, om daarna wat an-

mechanismen, *they gravitate automatically to the most efficient approach*, omdat ze niet worden beperkt door een bestaande organisatorische structuur. De manier van werken is heel sterk gebaseerd op meritocratie, dus niet op formele hiërarchie maar juist op technische competenties. Bij nieuwe ontwikkelingen zie je bijna altijd meerdere groepen die dat op verschillende manieren implementeren, waardoor een soort competitie ontstaat en uiteindelijk een van die implementaties wordt gekozen of verschillende implementaties worden samengevoegd. Je ziet nu dat bedrijven de principes van opensource in combinatie met Agile steeds meer toepassen.' Het uitwerken van een opensourcestrategie voor Intuit valt ook onder Bosch' werkzaamheden.

Voor de architectuur zijn de ontwikkelingen zo mogelijk nog ingrijpender. Die moet de applicatieontwikkelaars faciliteren – intern dan wel extern. 'De architectuur was traditioneel als een kathedraal: groots en ingewikkeld. Je moest echt een expert zijn om er wat mee te kunnen. Nu zie je dat de architectuur steeds meer wordt gebruikt om eenvoud te bereiken. In plaats van een heel grote, ingewikkelde integratiefase, die vaak aan het einde van een heel lang ontwikkelproces zit, zie je nu dat er een basisarchitectuur wordt neergelegd die met name compositionaliteit, terugwaartse compatibiliteit en eenvoud ondersteunt. Als de architectuur te groot en te complex is, wordt het te lastig om met kleine teams toch relevante contributies te doen. Amazon hanteert bijvoorbeeld de *three API rule*: geen team hoeft met meer dan drie Api's rekening te houden. We gaan naar een plek toe waarbij we de coördinatie niet meer via processen doen – processen worden door mensen uitgevoerd – maar juist door de architectuur.'

Jan Bosch verzorgt op dinsdag 20 oktober de keynote op het Practical Product Lines-event in Amsterdam. Voor informatie en inschrijven, zie www.practicalproductlines.org.