

Migratie naar Designer 6i

Uiteenlopende scenario's mogelijk

Het einde van het tijdperk van Designer/Developer 1 nadert nu echt met rasse schreden. Een groot aantal organisaties heeft nog gewacht, maar gaat nu beslissingen nemen voor het migreren van hun systemen. Het migreren van applicatiesystemen op basis van Designer/Developer versie 1 naar versie 6i is echter niet alleen een technische aangelegenheid.

Het migreren valt ook binnen de keuzes en visie op het snijvlak van informatietechnologie en organisatie. In dit artikel wordt, op basis van projectervaringen, ingegaan op drie belangrijke aspecten van deze migratie: de business- en technologydrivers voor de migratie; de migratie zelf en vervolgsenario's, en de voorbereiding van de migratie. Tezamen geven deze het IT management voldoende informatie om een weloverwogen keuze voor migratie te kunnen maken.

Business en technology drivers

We beginnen met een inventarisatie van migratietrajecten, waarbij Cap Gemini Ernst & Young betrokken is geweest. Die inventarisatie levert de volgende mogelijke 'drivers' op:

1. Geen support meer op de gebruikte versies

Designer/Developer release 6i is momenteel de enige versie waar Oracle volledig support op levert.

2. Afhankelijkheid van hardware en operating system

De bestaande infrastructuur waarop de applicatie kan draaien is afgeschreven, niet meer beschikbaar dan wel binnen korte termijn aan vernieuwing toe. Vernieuwing van de infrastructuur kan een probleem opleveren omdat de huidige Designer/Developer niet gecertificeerd is voor deze nieuwe infrastructuur.

3. Vernieuwing van de applicaties

Dit kan zowel vanuit functioneel als technisch oogpunt gewenst zijn. Een functionele aanleiding zou bijvoorbeeld kunnen zijn, als de insteek van de organisatie verandert. Er wordt bijvoorbeeld gewijzigd van een productgerichte naar een klantgerichte organisatie. Een technische aanleiding doet zich voor als bijvoorbeeld het inzetten van intranet-/internettechnologie gewenst is.

4. Gelijktrekken van alle gebruikte versies

De afgelopen jaren zijn meerdere applicaties op verschillende versies van Designer en Developer ontwikkeld. Het migreren van alle versies naar één versie, bijvoorbeeld Designer/Developer 6i, levert een optimalisatie voor het beheer op.

5. Upgrade naar database versie 8.1.7

De gewenste functionaliteit in een andere applicatie vereist een hogere versie in de database, 8.1.7 of wellicht zelfs hoger. Dit levert een probleem op bij de huidige Designer/Developer 1 applicatie, omdat deze door Oracle niet gecertificeerd is voor database versie 8.1.7.

6. Mogelijkheid tot versiebeheer

Een veelgehoorde reden voor overstap naar Designer 6i is de aanwezigheid van versiebeheer en de daarmee verwachte beheervoordelen.

Deze drivers hoeven niet allemaal te leiden tot een identieke aanpak voor migratie. Belangrijk is om de visie op applicatie-

Het einde van het Designer/Developer 1 tijdperk nadert nu echt met rasse schreden

ontwikkeling voor de komende drie jaar te bepalen en vanuit dat punt beslissingen te nemen. In de volgende paragrafen worden diverse scenario's besproken voor migratie, die al dan niet gecombineerd kunnen worden om aan de visie te voldoen.

Migratie scenario's

Is er een eenduidig migratiepad beschikbaar? Nee, helaas niet. Veel hangt af van wat we in de toekomst willen met de applica-

tie, hoe de applicatie nu is ingericht en wat de gedachten zijn over de toekomstige systeemontwikkeling. Het uitgangspunt is een Designer/Developer 1 omgeving. De volgende scenario's worden hieronder beschreven:

Minimale migratie inspanning voor de korte termijn

- Geen migratie
- Developer migratie

Continuering huidige systeemontwikkeling aanpak met Designer/Developer

- Migratie op basis van Developer
- Migratie op basis van Designer

Volledig nieuwbouw op basis van nieuw systeemontwikkelingsarchitectuur

- Nieuwbouw met Designer 6i
- Nieuwbouw met Designer 6i en JDeveloper
- Nieuwbouw met een nieuw softwareplatform (Java, .Net, Standaardpakket)

Minimale migratie inspanning

We hebben een applicatie waarvan we weten dat de functionaliteit de komende paar jaar minimale aanpassingen nodig heeft. Er is ook geen drastische wijziging in het aantal gebruikers en de applicatie draait stabiel op de hardware.

Scenario 1 - Geen migratie

In dit scenario blijft het applicatiesysteem gebaseerd op Designer/Developer versie 1 en op de betreffende databaseversie. In dit geval worden er geen kosten gemaakt voor migratie. Aan dit scenario zijn de volgende risico's verbonden:

- Geen support op de applicatie door Oracle (zie kader Designer/Developer support)
- Bij uitval van de hardware is geen uitwijk naar een nieuw hardware platform mogelijk. De huidige hardware en operating system (of besturingssystemen) moeten voor minimaal een tot twee jaar ondersteund en beschikbaar blijven.

Scenario 2 - Developer migratie

Bij dit scenario kunnen de migratiekosten voor de applicatie zoveel mogelijk beperkt blijven. In veel gevallen zal echter wel een investering in een nieuwe infrastructuur nodig zijn. In dit scenario worden de diverse forms en reports via een Developer conversie omgezet naar Developer 6i. Het verdere onderhoud van de applicatie na migratie zal in Developer worden uitgevoerd. Het eventuele gebruik van de nieuwe Designer versie zal beperkt blijven tot de documentatie in het 'uppercase' deel (ERD, business functions, en dergelijke). Door de uitrol van de

nieuwe run time omgeving bestaat de mogelijkheid de applicatie na migratie op basis van internet technologie aan te bieden. De beschikbaarheid van een web server is hiervoor noodzakelijk. De risico's op geen support en uitval van verouderde hardware worden met dit scenario vermeden.

Een logisch vervolg op scenario's 1 en 2 zou zijn nieuwbouw in een Designer/Developer 6i of JDeveloper omgeving of het opzetten van een volledig andere systeemontwikkelingsarchitectuur in een termijn van één tot twee jaar. Gebruik makend van deze nieuwe omgeving kan vervolgens de nieuwbouw van de te handhaven functionaliteit worden opgepakt.

Continuering huidige aanpak

In omgevingen waar al geruime tijd met Designer/Developer applicaties ontwikkeld en onderhouden worden, zullen de volgende twee scenario's logischerwijs snel op de voorgrond treden. Deze scenario's houden een continuering van de huidige systeem-

Welk migratiepad gekozen wordt, hangt af van wat we in de toekomst willen met de applicatie en de systeemontwikkeling

ontwikkeling in. De keuze voor het migratiescenario naar Designer/Developer 6i hangt af hoe er met honderd procent generatie en met bestaande templatestructuren zal worden omgegaan. We onderscheiden een migratie op basis van Developer, waarbij in de loop der tijd wordt overgegaan naar honderd procent generatie, en een migratie op basis van Designer, waarbij gestreefd wordt alle modules vanuit Designer te hergenereren.

Scenario 3 - Migratie op basis van Developer

In deze migratie wordt de applicatie gemigreerd met Developer. De repository-informatie van Designer 1 wordt naar Designer 6i gemigreerd. Gekozen wordt om alleen voor nieuwe functionaliteit te ontwikkelen met behulp van honderd procent generatie. Het onderhoud van de applicatie zal ook met Designer worden uitgevoerd. Bij het aanbrengen van wijzigingen aan bestaande schermen zal dus min of meer een uitgestelde migratie met behulp van Designer worden uitgevoerd voor de specifieke module. Dit zal het doorvoeren van wijzigingen aan nog niet gemigreerde modules vertragen.

Scenario 4 - Migratie op basis van Designer

Dit scenario sluit het best aan bij applicaties waar relatief veel wijzigingen op plaatsvinden. Omdat de migratie ingrijpend en bewerkelijk is, kost dit scenario duidelijk meer inspanning dan scenario 3. Voordeel van dit scenario is dat 100% generatie van alle modules in de applicatie, na de migratie behouden blijft. Na de migratie kan optimaal worden doorontwikkeld aan de gemigreerde applicatie, gebruikmakend van Designer 6i.

In dit scenario wordt de repository gemigreerd van Designer I naar Designer 6i en eventueel aangepast om weer volledig te kunnen genereren. De applicatie wordt daarna vanuit Designer volledig opnieuw gegenereerd in de nieuwe runtime versie. De voorbereiding van dit scenario bestaat uit het bepalen hoe de bestaande libraries en templates in Designer 6i worden gebruikt. Libraries en templates zullen deels aangepast moeten worden om in de nieuwe situatie optimaal gebruik te kunnen maken van Designer. Verder zal er bepaald moeten worden hoe de modulespecifieke functionaliteit in pll's, gebruikelijk in een Developer I omgeving, in de repository opgenomen gaat worden.

Templates en utilities

Het gebruik van templates en utilities verdient speciale aandacht bij de migratie van Designer/Developer. Bij scenario 3 en 4 kan gekozen worden om de bestaande templatestructuur en utilities (bedrijfsspecifiek of Headstart) één op één door te migreren naar Developer/Designer 6i en eventueel gebruik te laten maken van nieuwe mogelijkheden. Voor bedrijfsspecifieke templates en utilities kan dit een behoud van investeringen betekenen.

Een keuze voor een overgang naar Headstart 6i en CDM Ruleframe is de andere optie. Hierbij zal echter goed bekeken moeten worden welke impact dit heeft op de bestaande structuur van de applicatie en de standaarden en richtlijnen die worden gehanteerd. Wanneer er in de uitgangssituatie al volgens CDM wordt gewerkt, zal de overgang naar CDM en het implementeren van Rule Frame redelijk soepel kunnen verlopen. Wanneer er echter veel is geïnvesteerd in eigen standaarden en richtlijnen zal een overgang naar de nieuwe Headstart 6i, die sterk gekoppeld is aan CDM Rule frame, een traject met flinke impact tot gevolg hebben. De keuze om naast de bestaande (gemigreerde) omgeving een ontwikkelstraat in te richten die aansluit bij scenario 5 of 6 zou zo'n traject kunnen omzeilen.

Volledige nieuwbouw

Een keuze voor volledig nieuwbouw op basis van een nieuwe systeemontwikkelings-architectuur kan ingegeven worden door de wens om de applicaties opnieuw in te richten volgens een nieuwe architectuur (bijvoorbeeld 'van product- naar klantvisie'). De keuze kan ook worden ingegeven door de strategi-

sche overstap naar bijvoorbeeld een Java platform. We onderscheiden hierin drie mogelijk scenario's, die verschillen op basis van de te gebruiken technologie.

Scenario 5 - Nieuwbouw met Designer 6i

Er wordt verder gegaan met Oracle Designer als ontwikkelplatform. Omdat geen voorheen gebruikte standaarden en templates worden meegenomen kunnen extra functionaliteiten binnen Designer, eventueel in combinatie met Headstart 6i en CDM Rule Frame nu gebruikt worden. De keuze voor dit scenario geeft de mogelijkheid om een nieuwe applicatiearchitec-

Door de diversiteit aan implementaties, standaards en bibliotheken is in de praktijk een pilotmigratie noodzakelijk

tuur op te zetten op basis van een in de organisatie bekend ontwikkelplatform. De overstap van de organisatie van bestaande ontwikkelstandaards en richtlijnen naar Headstart 6i en CDM Rule Frame is op deze manier eenvoudiger te maken.

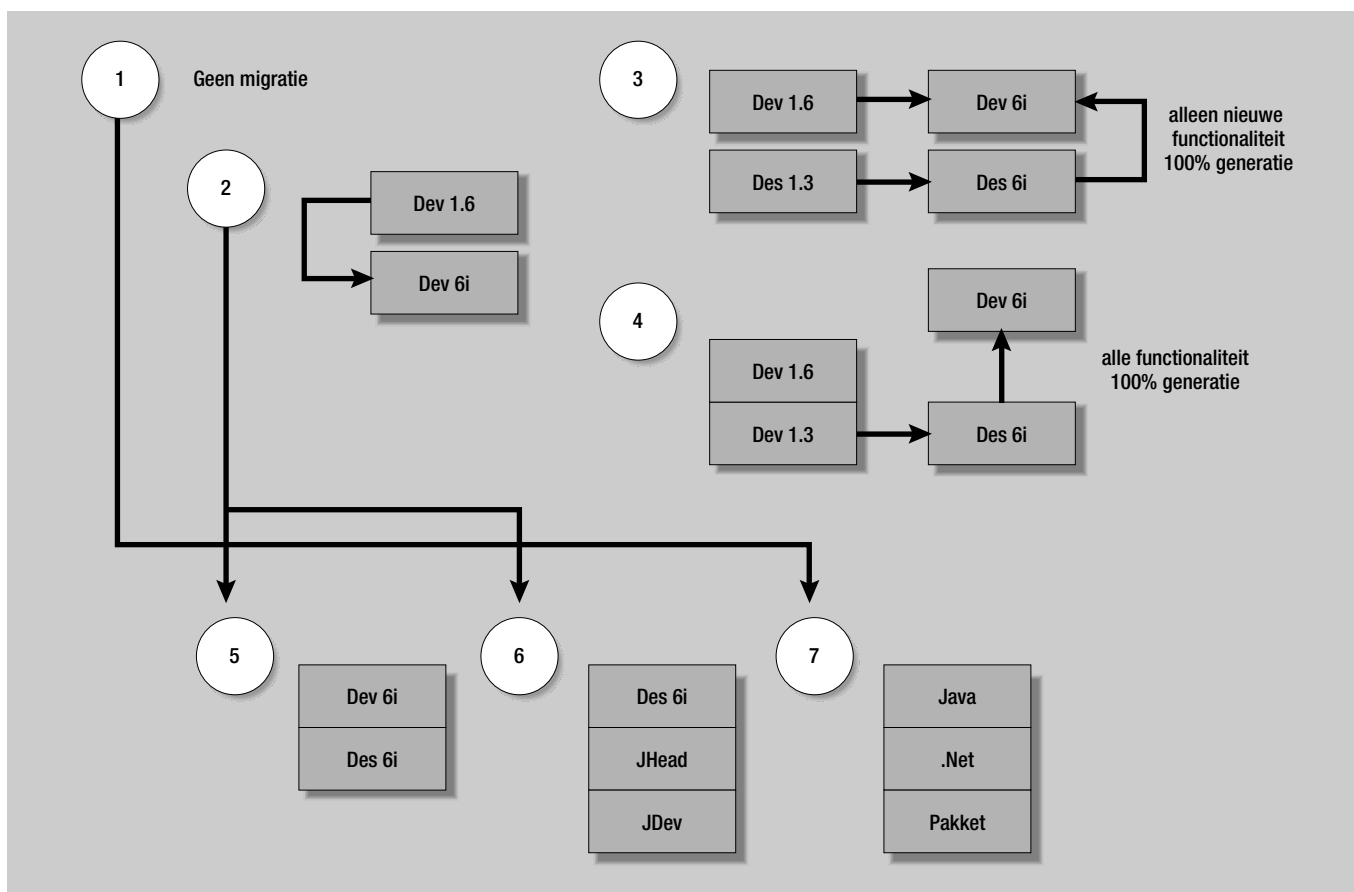
Scenario 6 - Nieuwbouw met Designer en JDeveloper

Met dit scenario wordt geen migratie van een reeds bestaande Designer/Developer omgeving bedoeld. We noemen het echter een scenario omdat het kan volgen op een bestaande situatie waar besloten wordt niet verder te ontwikkelen met behulp van Designer in combinatie met Developer. Dit scenario kan bijvoorbeeld volgen op scenario 1 of 2. Een belangrijk risico wat hierbij niet genegeerd mag worden is de nu nog beperkte ondersteuning die JHeadstart biedt voor de generatie van Java applicaties vanuit de Designer Repository. Dit scenario bevindt zich op dit moment in een pioniersfase.

Overigens kan bij scenario 4 en 5 in een vervolgfase eventueel gebruik worden gemaakt van JHeadstart met de generatie vanuit Designer. Dit om bijvoorbeeld voor een bepaald deel van een backofficesysteem een 'light weight' intranetmodule te genereren.

Scenario 7 - Nieuwbouw met een nieuw softwareplatform (Java, .Net of standaardpakketten)

Evenals scenario 6 is dit geen migratie. Er wordt afscheid genomen van de bekende ontwikkeling met Oracle Forms/Reports als productieomgeving. Een overstap naar een Java richting met



Figuur 1. Bij keuze voor scenario 1 of 2, minimale migratie inspanning voor de korte termijn, zijn de scenario's 5, 6 of 7 de logische vervolgstap op een termijn van een à twee jaar. De scenario's 3 en 4 houden vast aan de gangbare ontwikkeling van applicaties gebruikmakend van Designer en Developer. De keuze voor een van deze scenario's houdt niet in dat een ontwikkeling naar 5, 6 of 7 niet mogelijk is. Een dergelijke ontwikkeling kan echter beter als een losstaand traject beschouwd worden. Niet vermeld in de figuur is het doorgroeien vanuit scenario 4 naar scenario 6. Dit traject, met behulp van JHeadstart Designer Generator, is momenteel in ontwikkeling. Indien er voldoende duidelijkheid bestaat hoe er binnen dit scenario omgegaan zal worden met verschillende aspecten van systeemontwikkeling, is een aanpak gekoppeld aan scenario 4 te overwegen.

gebruik van JDeveloper is een richting die binnen de Oracle scope blijft. Het kiezen voor een totaal andere omgeving, bijvoorbeeld .Net, zou echter ook kunnen. Hierbij moet de kracht van de Oracle database omgeving echter niet uit het oog worden verloren. Tenslotte is een overgang naar het gebruik van standaardpakketten wellicht ook een mogelijkheid.

In figuur 1 zijn de onderlinge relaties tussen te onderscheiden scenario's schematisch weergegeven.

Van client/server naar Internet computing

Voor de deployment van een Forms en Reports applicatie op het intranet is een migratie naar in ieder geval Developer 6i aan te raden, aangezien dit de versie is die deze deployment het beste ondersteunt. Er dient extra rekening te worden gehouden met diverse aandachtspunten. Hieronder een niet uitputtende lijst:

- Inrichting van een Internet Applicatie Server met Forms en Reports Server

- Bandbreedte en latency van het interne netwerk
- Capaciteit van de aanwezige Personal Computers in verband met het gebruik van Java applets
- Indien er nu een client/server architectuur is, dient een andere versiebeheer procedure en uitrolmechanisme opgezet te worden
- Aanroepen in de applicatiemodules naar het Operating Systeem vinden plaats op de applicatieserver in plaats van de Cliënt. Deze aanroepen dienen eventueel herschreven te worden
- De output van Reports kan in Pdf, Html en XML gegenereerd worden
- Bij gebruik van Windows Help functionaliteit, dient deze herschreven te worden naar HTML pagina's. Een andere optie is om de meegemigreerde Help-informatie vanuit de repository opnieuw te genereren, maar nu naar een HTML structuur

Vorbereiding migratie

Omdat de migratie invloed heeft op keuzes en visie op het vlak van informatietechnologie en organisatie, is een grondige voor-

bereiding noodzakelijk. Als eerste is het van belang uit te vinden welk migratiescenario, en eventuele gelijktijdige en vervolgsenario's, gebruikt gaan worden. De betrokken groepen in dit proces zijn behalve Beheer, Architecten, Informatieanalisten en de Ontwikkelafdeling ook het Management en Gebruikers. Nadat de strategische keuzes voor migratie en de toekomstige systeemontwikkeling zijn uitgedacht kan de voorbereiding getroffen worden voor de migratie. In de praktijk is duidelijk geworden dat het uitvoeren van een pilotmigratie hierin een noodzaak is. Door de diversiteit aan implementatiemogelijkheden, gebruik van standaarden en bibliotheken bestaat er namelijk helaas geen eenduidige migratiestrategie. In zo'n pilot wordt een beperkt deel van de applicatie gemigreerd in een losstaande omgeving volgens het gekozen migratiescenario. Tijdens deze pilot kunnen specifieke aandachtspunten aan het licht komen, waaraan in de totale migratie extra aandacht besteed kan worden. Een voorbeeld hiervan is een situatie waar met een aangepaste Headstart versie wordt gewerkt. In zo'n geval kan het zijn dat een overgang naar de nieuwe Headstart 6i extra werkzaamheden per module moeten worden uitgevoerd tijdens de migratie. Dit om de applicatie te kunnen blijven onderhouden met Designer. De tijd die hiermee gemoeid is en de wijze waarop dit uitgevoerd moet worden zal in het plan specifieke aandacht verdienen.

Met alle opgedane kennis kan hierna een plan van aanpak voor de migratie opgesteld worden. Standaardonderdelen in het plan van aanpak voor migratiescenario's 3 en 4 zijn de volgende:

- Inrichten nieuwe omgeving naast de bestaande
- Opschonen bestaande omgeving
- Een Designer 6.0 repository beschikbaar maken om de migratie van de Designer applicatiesystemen naar 6i te kunnen uitvoeren
- Impactanalyse van de templatestructuur in de bestaande applicatie
- Categorisering van de modules naar complexiteit om een betere raming van de benodigde tijd per module te kunnen maken
- Voldoende aandacht besteden aan het testen van de applicatie
- Beschikbaarheid van mensen met kennis van de applicatie voor participatie in het migratietraject
- Kennisopbouw van de nieuwe ontwikkelomgeving en architectuur.
- Voorbereiding van het testtraject

Conclusie

Met het nu echt eindigen van de levensloop van Designer/Developer I komt voor bedrijven het moment om een keuze te nemen die verder gaat dan alleen de migratie van de huidige Developer I applicatie. Een keuze van de toekomstige systeemontwikkelingsarchitectuur is gelijk onderdeel van dit traject. In het kort zijn de volgende vervolgttrajecten mogelijk:

- Minimale migratie inspanning voor de korte termijn
- Continuering huidige systeemontwikkeling aanpak met Designer/Developer

- Volledig nieuwbouw op basis van nieuw systeemontwikkelingsarchitectuur

In omgevingen waar al geruime tijd met Designer/Developer applicaties ontwikkeld en onderhouden worden, zullen de migratiescenario's op basis van Developer en Designer (scenario's 3 en 4) logischerwijs snel op de voorgrond treden. Omdat de migratie invloed heeft op keuzes en visie op het vlak van informatietechnologie en organisatie, is een grondige voorbereiding noodzakelijk.

Designer/Developer Support

Oracle verdeelt zijn support grofweg in twee delen,

- Error Correction Support (ECS) en
- Extended Assistance Support (EAS).

Bij ECS worden bugs in producten middels patches verholpen, worden de producten gecertificeerd voor nieuwe versies van operating systemen en kan, afhankelijk van het contract, een beroep op experts worden gedaan.

EAS volgt na de beëindiging van ECS en geeft alleen nog telefonische en elektronische support op vragen, mogelijke workarounds en advisering bij migratie naar gesupporte versies. In onderstaande tabel is huidige supportinformatie samengevat voor een aantal producten (bron: Metalink januari 2002).

Support	ECS	EAS
Designer I	31-12-2000	31-12-2003 (*)
Developer I	31-12-2000	31-12-2003 (*)
Designer 6.0	31-12-2001	31-12-2004
Developer 6.0	30-06-2001	30-06-2004
Dbase 7.3.4	31-12-2000	31-12-2002 (**)

* alleen voor bestaande support contracten

** voor klanten die vóór de datum niet kunnen upgraden (extended maintenance support)

Joost van Cappellen (joost.van.cappellen@cgey.nl),

Joost van der Vlies (joost.vander.vlies@cgey.nl) en

Léon Smiers (leon.smiers@cgey.nl)

zijn werkzaam als senior consultants bij Cap Gemini Ernst & Young.