



thema

Herkent u de volgende uitspraken?

‘Er is een applicatie ontwikkeld, maar we krijgen hem niet stabiel.’

‘In de testomgeving doet de applicatie het prima, alleen in productie wil het niet werken.’

‘Het uitbreiden van functionaliteit aan een applicatie zou toch niet zo’n probleem mogen zijn, maar er gaat altijd wel wat mis en kost altijd meer inspanning en downtime dan gepland.’

‘Eigenlijk moet die server nodig opnieuw worden ingericht, maar die applicatie krijgen we dan vast nooit meer aan de praat’.

Waarom applicaties niet werken

Onvermoede achilleshiel blootgelegd

De applicatie is uitgebreid getest in de ontwikkelomgeving en alle gevonden fouten zijn opgelost. Verder is de applicatie goed gedocumenteerd, zowel op functioneel niveau als op technisch niveau. Waar komen deze implementatie- en beheersproblemen dan vandaan?

Bijna iedere applicatiebeheerder of applicatieontwikkelaar herkent tenminste één van bovenstaande situaties. Meestal zijn ze het resultaat van een project dat prima verliep tot er onverwacht problemen ontstonden tijdens de implementatie. De problemen werden snel opgelost, maar na oplevering van de applicatie doken er opnieuw problemen op. Met de nodige inspanning werden ook deze problemen (zo goed mogelijk) opgelost. Nieuwe aanpassingen resulteren nu echter steeds vaker in nieuwe, moeilijk te achterhalen en op te lossen problemen, waardoor er inmiddels niet veel medewerkers meer zijn die nog iets durven te wijzigen aan de huidige implementatie.

Dit soort situaties komen vaak voor en niet alleen bij de ontwikkeling van maatwerk-oplossingen, maar ook bij ‘off the shelf’ pakket implementaties. Uit ervaring blijkt dat vooral kleine en middel grote projecten gevoelig zijn voor dit soort problemen.

Ook al is er goed gedocumenteerd en is in de ontwikkelomgeving uitgebreid getest, wat vrijwel nooit gebeurd is het documenteren van de manier waarop de applicatie moet worden geïnstalleerd en het testen van de beschreven installatie procedure.

Tijdens een ontwikkeltraject worden er zeer veel wijzigingen aangebracht in een ontwikkelomgeving. Niet alleen in de code, maar ook in de omgeving waarin de applicatie wordt ontwikkeld en getest. Er komen voortdurend patches en nieuwe versies uit voor de ontwikkelomgeving, het operating system en ondersteunende system libraries (XML engines, ODBC drivers, et cetera). Deze worden op verschillende manieren en op verschillende momenten tijdens het ontwikkeltraject geïnstalleerd. Indien men niet precies bij heeft gehouden wat allemaal is geïnstalleerd en wanneer en op welke manier, dan is de kans groot dat er tijdens de implementatie problemen ontstaan.

UITEENLOPENDE OMGEVINGEN Dit is echter pas het eerste deel van het probleem. De implementatie problemen worden meestal nog wel (gedeeltelijk) opgelost. Gewoon door vergeten patches en updates te installeren, net zo lang totdat de gevonden implementatie problemen zijn verholpen. Het gevolg is echter dat er verschillen zijn ontstaan tussen de fysieke inrichting van de ontwikkelomgeving en die van de productieomgeving.

Weinig applicaties zijn honderd procent foutvrij als ze in productie worden genomen. Dit betekent dat er na de implementatie nog geregeld aanpassingen worden gedaan in zowel de ontwikkelomgeving als de productieomgeving. Het niet documenteren van de updates

heeft tot gevolg dat het overzicht over wat precies wel en niet geïnstalleerd is, verder verloren gaat – fouten worden niet meer reproduceerbaar. Het verschil tussen de ontwikkelomgeving en de productieomgeving wordt hiermee steeds groter; het project geraakt in een vicieuze cirkel. Eigenlijk zou men de productieomgeving opnieuw willen installeren, maar dat durft men niet aan uit vrees dat de eerdere implementatie problemen zich allemaal opnieuw zullen voordoen. Er ontstaat een patstelling. Uit ervaring blijkt dat gebrekkige kwaliteit en stabiliteit van een applicatie niet (alleen) moet worden gezocht in de kwaliteit van de code, maar vaak meer in de manier waarop er met deze code wordt om gegaan.

Het hiervoor beschrevene leidt tot stevige financiële consequenties.

- Er worden extra projectkosten gemaakt en eventueel loopt het project uit door de extra inspanningen die nodig zijn voor het oplossen van de implementatie problemen.
- De beheerskosten voor de applicatie liggen veel hoger dan geraamd doordat er meer problemen zijn en het langer duurt om de problemen te lokaliseren en op te lossen.
- De effectiviteit van de applicatie wordt verminderd doordat gebruikers en beheerders gefrustreerd raken door de problemen en daarom ze minder gebruik te maken van de applicatie.
- De levensduur van de applicatie wordt verkort doordat het moeilijk of bijna onmogelijk wordt om wijzigingen aan te brengen of de applicatie uit te breiden.

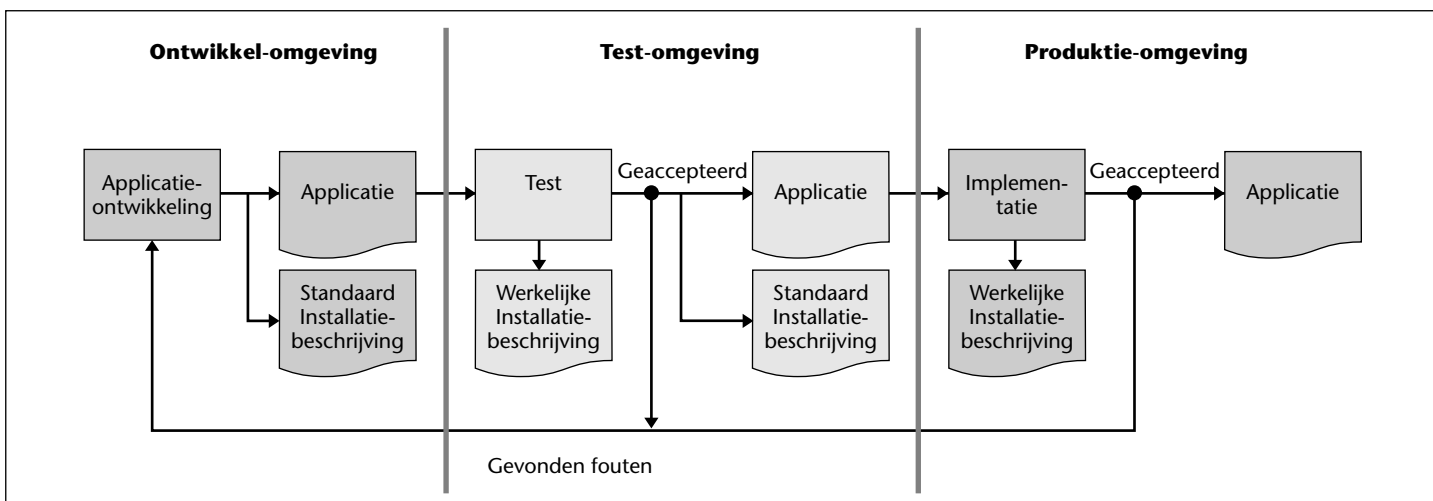
PIJNPUNT? De kern van het probleem ligt in het gebrek aan documenteren en testen van de installatie procedures. Tijdens een implementatie dient de inrichting van een applicatie zoals deze draait in de ontwik-

kelomgeving, te worden gereproduceerd in de productie omgeving. Hierbij moeten alle onderdelen van die inrichting die essentieel zijn voor de applicatie zijn meegenomen. Tot de essentiële onderdelen voor een applicatie behoren niet alleen de verschillende stukken code van die applicatie, maar ook het operating system

Voortdurend komen patches en nieuwe versies uit voor ontwikkelomgeving, OS en ondersteunende system libraries

zelf en de ondersteunende system libraries, zoals ODBC drivers en XML engines. Dit betekent dat er een installatiebeschrijving moet zijn waarin al deze onderdelen zijn opgenomen en waarin staat beschreven in welke volgorde zij moeten worden geïnstalleerd. Bovendien moet voor elk onderdeel zijn beschreven hoe het moet worden geïnstalleerd en vooral niet te vergeten waarmee (Windows installer, set up versies). De installatie van een net even iets andere versie van een bepaalde system library kan leiden tot vervelende en zeer moeilijk te traceren problemen.

Bovenstaande geldt niet alleen voor de implementatie van de eigenlijke applicatie, maar ook voor iedere update die later wordt uitgevoerd. Verder is een omgeving noodzakelijk waarin de installatie procedures kunnen worden getest. Hiervoor zijn de ontwikkelomgeving zelf en de productieomgeving ongeschikt. Indien er een tweede testomgeving beschikbaar is kan deze hiervoor worden gebruikt, zo niet dan zal deze moeten worden gerealiseerd. Ook voor de kleine projecten. Dit hoeft overigens niet duur te zijn, de standaard denkfout die vaak wordt gemaakt. Het gaat om het proces waar-



FIGUUR 1. Het implementeren van een applicatie in een ontwikkel-, testomgeving en uiteindelijk in de productieomgeving.

bij een omgeving ook op een desktop kan draaien. Het testen van de installatie procedure kan overigens het beste worden gecombineerd met de systeemtest van de applicatie.

PREVENTIE De oorzaak voor het ontbreken van beschrijvingen voor installaties en installatie procedures is gelegen in het feit dat het documenteren van installatie procedures over het algemeen niet als onderdeel wordt gezien van het eigenlijke ontwikkeltraject. Als het al wordt gezien als een projectactiviteit dan wordt het meestal gepland tijdens de implementatiefase als onderdeel van het maken van de overdrachtsdocumentatie. Dit in combinatie met het feit dat het documenteren van installatie procedures vaak als vervelend werk wordt beschouwd, zorgt er voor dat er niet wordt gestart met beschrijven tot het te laat is. Achteraf beschrijven hoe een omgeving daadwerkelijk is ingericht is erg lastig en blijft meestal beperkt tot een grove

tatie die nodig is om tot een installatie te komen, wordt vastgelegd in de repository.

- *Software CD's*

CD's zijn meestal onpraktisch om te kopiëren naar een repository, maar het is wel belangrijk dat binnen het project altijd dezelfde (versies van) CD's worden gebruikt. Maak daarom project kopieën van alle CD's die in het project nodig zijn en bewaar deze in een kast in de (buurt van) de project ruimte. Voorzie alle CD's van een code en eventueel versie nummers zodat het makkelijk is om in de documentatie aan te geven welke CD wanneer moet worden gebruikt, c.q. is gebruikt.

- *Oude versies*

Als er nieuwe versies van software binnenkomen, is het belangrijk dat de oude versies niet worden weggegooid. Nieuwe versies van software kunnen gewoon worden voorzien van een hoger versie nummer zodat het duidelijk is wanneer welke versie is gebruikt. De oude versies van de software moeten worden bewaard om de bestaande installaties te kunnen reproduceren. Als in uw organisatie de project-CD's regelmatig zoekraken, kan er een set 'master kopieën' worden gemaakt die door bijvoorbeeld een secretariaat worden bewaard.

- *Installatie documentatie*

Het centraal opslaan van software en documenten is niet datgene waar de meeste project medewerkers tegen op zullen kijken. Het documenteren van installatie procedures wel. Er zijn twee soorten installatie documenten te onderkennen, te weten de documentatie die beschrijft hoe een applicatie zou moeten worden geïnstalleerd (*from scratch*) en de documentatie die beschrijft hoe een bepaalde installatie daadwerkelijk is uitgevoerd.

In de documentatie van de daadwerkelijke installatie moet de volledige installatie van de server worden beschreven plus alle updates die er later op zijn uitgevoerd. Het document begint dus met de beschrijving van de installatie van het operating system en eindigt met de laatste software update. Registreer de datum en tijd bij iedere update van het document zo dat duidelijk is wanneer welke updates zijn toegepast.

Er is tenminste één installatie beschrijving nodig die als handleiding kan worden gebruikt voor de implementatie van de applicatie in productie en de testomgevingen. Hiervoor kan het beste een apart installatie document worden gecreëerd, waarbij de installatie specifieke onderdelen (server naam, ip-adres, user-id's, passwords, et cetera) zijn vervangen door placeholders. Deze placeholder functioneren dan als een soort parameters. Een dergelijke beschrijving kan eenvoudig wor-

Vrijwel nooit wordt de wijze van installatie en het testen van de procedure goed gedocumenteerd

benadering van de werkelijkheid. Om dit te voorkomen is het belangrijk dat het documenteren van installaties en installatie procedures als aparte activiteit wordt opgenomen in het project en dat de documenten als op te leveren producten worden opgenomen aan het einde van iedere activiteit waarbij installaties van worden gedaan.

Het bovenstaande klinkt als 'verschrikkelijk veel werk en erg ingewikkeld', maar dat hoeft het in de praktijk helemaal niet te zijn.

- *Repository's*

Allereerst moet er een vaste plek worden afgesproken waar alle (installatie) documenten en alle installatie software wordt opgeslagen. Een dergelijke repository kan gewoon een directory op een centrale fileservers zijn, maar kan ook met geavanceerde software (SourceSafe, PVCS en ClearCase bijvoorbeeld) worden gerealiseerd. Belangrijk is dat er een vorm van versiebeheer wordt gecreëerd zodat project medewerkers niet per ongeluk de wijzigingen van elkaar overschrijven. Dit kan worden gerealiseerd door afspraken te maken of met behulp van een tool. Indien er reeds een repository is voor de ontwikkelde code van de applicatie, kan de installatie software en documentatie hier aan worden toegevoegd. Zorg er voor dat *alle* software en documen-

den gecreëerd door de beschrijving van de daadwerkelijke installatie van ontwikkelomgeving te kopiëren en aan te passen. Het creëren van beschrijvingen van nieuwe installaties is hierna niets meer dan kopiëren en aanpassen.

Ga bij het maken van de documentatie uit van het 'schoonmoeder principe'. Probeer de installatie procedures zodanig te beschrijven dat je schoonmoeder in staat is om met behulp van de installatie beschrijving en de juiste software, de installatie te reproduceren. Ga er niet van uit dat de collega's (of je schoonmoeder) wel begrijpen wat je bedoelt. Een beroemd Engels spreekwoord zegt niet voor niets: 'The assumption is the mother of all fuck-ups'. Een goede installatie beschrijving zorgt er voor dat vrijwel iedereen een installatie kan uitvoeren. Dit zorgt er weer voor dat een project minder afhankelijk wordt van (dure) specialisten op dit gebied.

Indien men denkt lang gebruik te maken van een applicatie is het verstandig om niet alleen vast te leggen hoe een bepaalde installatie is uitgevoerd, maar ook waarom bepaalde installatie keuzes zijn gemaakt. Dit maakt het eenvoudiger om later aanpassingen in de installatiewijze aan te brengen, doordat het waarom is vastgelegd.

- *Scripts*

De hoeveelheid handelingen die moeten worden gedocumenteerd en daarmee de daarvoor benodigde inspanning, kan worden beperkt door zo veel mogelijk te werken met scripts. Bijvoorbeeld; een unattended installation file kan een groot deel van de handmatige installatie handelingen van een Windows2000 server vervangen waardoor deze handelingen ook niet meer hoeven te worden gedocumenteerd. Dit vermindert tevens de kans op installatie fouten tijdens de implementatie. De unattended installation file wordt verder gewoon in de repository opgeslagen.

- *Releases*

Indien we te maken hebben met een applicatie die bestaat uit grote hoeveelheden bestanden die vaak wijzigen, zoals bijvoorbeeld een website, is het handig om te werken met releases. Indien men een versie van de site wil opleveren naar de testomgeving, kopieert men alle bestanden die tot de site behoren, inclusief eventuele installatie scripts, naar een zipfile. Deze zipfile krijgt een naam met daarin een versie nummer en wordt naar de repository gekopieerd. De repository is zoals afgesproken de bron van alle installatie software en op deze manier kan men later altijd weer beschikken over die specifieke combinatie van files indien dit nodig mocht zijn. Door de website versie als parameter op te nemen in de installatie beschrijving van de website, hoeft niet

voor iedere nieuwe release de installatie beschrijving te worden aangepast.

UITDAGING Uit ervaring is gebleken dat er meer nodig is voor stabiele en beheersbare applicatie dan het documenteren van de functionaliteit en het testen van de code. Net zo belangrijk is het beschrijven van de manier waarop de omgeving moet worden gecreëerd

Een beroemd Engels spreekwoord zegt niet voor niets: 'The assumption is the mother of all fuck-ups'

waarin deze applicatie moet gaan functioneren. 'De ketting breekt bij de zwakste schakel', daarom is het belangrijk dat de kwaliteit van alle schakels wordt geborgd. Door het beschrijven en testen van de installatie procedures onderdeel te maken van het ontwikkeltraject kan dit worden gewaarborgd.

Dit kost tijd en energie. Bovendien is documenteren niet de favoriete bezigheid van de meeste ontwikkelaars. Door de installatieprocedures efficiënt in elkaar te zetten kan de kwaliteit van de installatie echter enorm worden verbeterd, terwijl de benodigde documentatie inspanning wordt geminimaliseerd. Door de hogere kwaliteit zal dit niet tot extra project kosten leiden omdat er minder problemen zullen zijn en de gevonden problemen sneller zullen worden opgelost. De uitdaging zal vooral liggen in het overtuigen van alle projectmedewerkers van de noodzaak van goede installatiedocumentatie.

Gérard van der Burg en Arjen Karten zijn werkzaam als senior infra-engineers bij Warp11 de webpractice van Cap Gemini Ernst & Young.