

Het in productie en beheer nemen van een J2EE-applicatie levert regelmatig discussie op tussen beheer- en ontwikkelafdelingen. Beheerorganisaties worden vaak pas laat bij projecten betrokken en weten ook niet altijd hun beheereisen gestructureerd te formuleren. Ontwikkelprojecten aan de andere kant richten zich primair op functionaliteit voor de eindgebruiker en hebben vaak te weinig aandacht voor de exploitatie-aspecten van hun applicaties. Het ISO 9126-model voor software-productkwaliteit biedt een goede kapstok voor het definiëren van standaard intakecriteria voor J2EE-applicaties.

thema

Is uw J2EE applicatie klaar voor beheer?

ISO-model voor software-productkwaliteit

Het Extended ISO 9126 model definieert een zestal *eigenschappen* voor de kwaliteit van softwareproducten. Elk van deze eigenschappen kent weer een aantal *attributen* die samen de betreffende eigenschap definiëren.

Hoe theoretisch het ISO model ook mag lijken, het blijkt in de praktijk een handige kapstok voor het definiëren van intakecriteria voor beheer van J2EE applicaties. Omdat het model ook alle functionele- en bruikbaarheidsaspecten van software afdekt, zijn vanuit een beheerperspectief niet alle eigenschappen en attributen even relevant. Het zwaartepunt voor beheer ligt namelijk bij de eigenschappen efficiëntie, onderhoudbaarheid, portabiliteit en betrouwbaarheid. Het belang van de kwaliteitseigenschappen verschilt per organisatie. Zo zal een financiële instelling de veiligheid van een applicatie zwaarder laten wegen dan een uitgever. Ook kijken verschillende rollen zoals ontwikkelaar en beheerder ieder op hun manier naar de eigenschappen. Het is daarom van belang de beheereisen zo goed mogelijk vast te leggen. Ze moeten goed aansluiten bij de doelstelling van de onderneming en voor alle belanghebbenden helder zijn. Voor een dergelijke vastlegging is een standaardnotatie nodig. Tom Gilb beschrijft in zijn boek *Competitive Engineering*² een complete taal voor het gestructureerd vastleggen van eisen. Deze taal heet *Planguage*, wat is afgeleid van de woorden 'plan' en 'language'. Een sterk punt van de aanpak van Tom Gilb is dat hij dwingt eisen te kwantificeren. Alle eisen moeten daarbij worden voorzien van een schaal en een meetin-

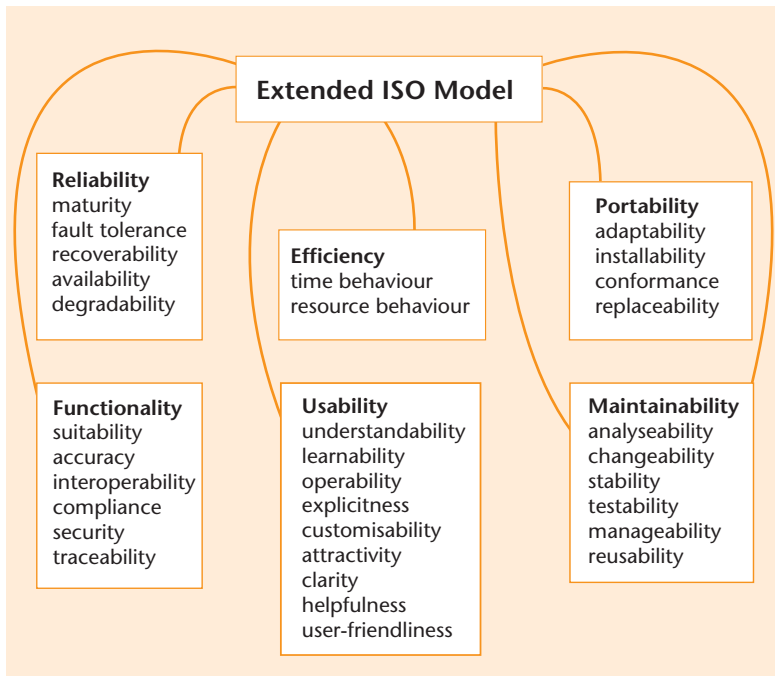
strument waartegen zij worden beoordeeld. Planguage kent ondermeer de volgende elementen voor het beschrijven van eisen:

Essentie	Een korte beschrijving die de essentie van de eis weergeeft.
Schaal	Een maatstaf die de meetbare eenheden specificeert voor een attribuut of eis.
Meetinstrument	Een praktische methode voor het meten en testen van een attribuut met een gedefiniëerde schaal.
Doel	Een gekwantificeerd doel. De consequentie van een doel is dat er een commitment is of komt om het doel te behalen.
Wens	Een gewenst of zelfs gedroomd doel. Een wens hoeft niet altijd realistisch of haalbaar te zijn maar geeft wel een streefrichting aan.

TABEL 1.

INTAKECRITERIA VOOR J2EE APPLICATIES In het vervolg van dit artikel zullen we *Planguage* gebruiken om een aantal van de kwaliteitsattributen uit het extended ISO-model te voorzien van een gekwantificeerde eis.

EFFICIËNTIE Deze verzameling eigenschappen geeft het verband aan tussen de prestaties (performance) van een applicatie en de hoeveelheid middelen die daarvoor nodig zijn.



FIGUUR 1. Extended ISO 9126 Model van Serc/Quint¹.

Tijdsbeslag. De performance van een applicatie is van grote invloed op het gebruikersgemak van een applicatie. Beheerders hebben echter ook belang bij een goed presterende applicatie, omdat een trage applicatie de stabiliteit van een applicatieserver kan bedreigen. Net als bij de kassa's in een supermarkt probeert een applicatieserver zo snel mogelijk het binnenkomende werk af te handelen. Als het afhandelen van bepaalde requests lang duurt, kunnen er lange wachtrijen ontstaan en kan het systeem instabiel worden. Het goed definiëren van eisen ten aanzien het tijdsbeslag dat een applicatie op de beschikbare middelen legt is daarom essentieel. Voorbeelden van vragen die gesteld kunnen worden zijn: Zijn er (deel)functionaliteiten die lang duren en kunnen zij andere functionaliteiten negatief beïnvloeden? Is er gedacht aan scheduling/batch-plan-

Eis	efficiëntie.tijdsbeslag.responsetijd.mywebservice
Essentie	De snelheid waarmee de applicatie een aanroep van mywebservice kan afhandelen.
Schaal	De frequentieverdeling van de requestduur over vooraf gedefinieerde klassen van responsetijden.
Meetinstrument	Een loadtest met behulp van Jmeter met het [testscript].
Doel	95% < 1 sec., 99% < 3 sec.
Wens	100% < 1 sec.
[testscript]	<Een definitie van het te gebruiken Jmeter testscript inclusief specificatie van de workload>

TABEL 2.

ning en is deze te beïnvloeden? Worden er performance-testen uitgevoerd? Al deze vragen moeten vertaald worden naar een meetbare eis. Voor de performance van een denkbeeldige webservice zouden we bijvoorbeeld de volgende eis kunnen definiëren:

Middelenbeslag. J2EE-applicatieservers hosten vaak meerdere applicaties. Het is daarom van belang de impact te kunnen inschatten van een applicatie op andere applicaties en processen die op dezelfde infrastructuur draaien. Betrek daarom de beheerorganisatie bij load- en regressietesten van de applicatie. Dit levert inzicht in benodigde resources zoals het CPU- en geheugengebruik. Formeel gedefinieerd levert dit bijvoorbeeld de volgende eis op:

Eis	efficiëntie.middelenbeslag.memory
Essentie	Het beslag dat de applicatie legt op het geheugen.
Schaal	MB.
Meetinstrument	Max heap size setting JVM tijdens performance testen.
Doel	<= 512 MB
Wens	<= 256 MB

TABEL 3.

ONDERHOUDBAARHEID Deze eigenschap is een maatstaf voor de hoeveelheid werk die nodig is om de applicatie te beheren en wijzigingen aan te brengen.

Analyseerbaarheid. De analyseerbaarheid van een applicatie wordt sterk bepaald door de aanwezigheid en kwaliteit van de documentatie. Het gaat daarbij niet alleen om ontwerpen of documentatie van de code, maar ook om documentatie voor beheerders. Denk daarbij aan een beschrijving van alle foutmeldingen inclusief de mogelijke oorzaak daarvan en oplossingen. Er kunnen tal van eisen worden gesteld ten aanzien van de op te leveren documentatie. Voorbeelden voor eisen aan de JavaDoc zijn bijvoorbeeld:

Eis	onderhoudbaarheid.analyseerbaarheid.javadoc.compleetheid
Essentie	De compleetheid van de JavaDoc documentatie.
Schaal	% van items waar JavaDoc commentaar mist.
Meetinstrument	Sun DocCheck tool.
Doel	< 10%
Wens	0%

TABEL 4.

Natuurlijk zegt de compleetheid van de JavaDoc niet alles. Ook de kwaliteit van de documentatie moet beoordeeld worden. Ook dit kan in een meetbare eis worden vastgelegd:

Eis	onderhoudbaarheid.analyseerbaarheid.java-doc.kwaliteit.interfaces
Essentie	De kwaliteit van de (publieke) interfaces van componenten.
Schaal	% JavaDoc commentaar met inspectie bevindingen.
Meetinstrument	Manuele [inspectie]. Steekproef op 10% van de interfaces.
Doel	< 10%
Wens	0%
[inspectie]	- Controle op aanwezigheid van beschrijvingen van pre- en postcondities. - duidelijk beschreven wat het component doet/biedt.

TABEL 5.

Wijzigbaarheid. Wijzigbaarheid beantwoordt de vraag of er snel wijzigingen in de applicatie kunnen worden aangebracht. Relevante vragen daarbij zijn: Kunnen er snel patches worden aangebracht? Is er een standaard buildscript zodat de applicatie in één keer gebouwd kan worden? Is er een versiebeheersysteem waaruit altijd de juiste versie van de sources geconstrueerd kunnen worden? Zijn er geautomatiseerde regressietests, is er een snelle (automatische) wijze van deployen van de J2EE-applicatie?

Eis	onderhoudbaarheid.wijzigbaarheid.doorlooptijd
Essentie	De tijd die nodig is om een wijziging inclusief complete regressietest door te voeren in productie.
Schaal	Tijd in uren.
Meetinstrument	Simulatie oefening.
Doel	< 24 uur
Wens	< 4 uur

TABEL 6.

Stabiliteit. Een belangrijke indicator voor de stabiliteit van een applicatie is het aantal geconstateerde fouten per release. Voor deze indicator kunnen eenvoudig eisen worden geformuleerd. Bijvoorbeeld het maximaal aantal incidenten in de eerste week na invoering.

Testbaarheid. Testbaarheid gaat over de inspanning om de applicatie te valideren. We hebben het dan vanzelfsprekend over unit testen, regressietesten en accep-

tatietesten van nieuwe functionaliteit. Voor de beheerder is echter ook een verificatietest na installatie van belang. Is de applicatie ook testbaar in productie? Vaak moeten hiervoor monitoringscripts worden gecreëerd die gebruikers simuleren. Testbaarheid betekent ook dat er ontwikkel- en testomgevingen zijn die volledig vergelijkbaar zijn met de productieomgeving, zowel qua hardware als qua software. Het is van essentieel belang dat de gebruikte JDK, applicatieserver en fixpacks op alle omgevingen gelijk worden gehouden.

Beheerbaarheid. De beheerbaarheid van een applicatie lijkt vaak een vaag begrip en er worden dan ook zelden concrete eisen aan gesteld. Toch is dit begrip vrij eenvoudig te kwantificeren:

Natuurlijk is bovenstaande eis vrij abstract en geeft

Eis	Onderhoudbaarheid.beheerbaarheid
Essentie	De inspanning die nodig is om de applicatie in de lucht te houden.
Schaal	De ratio van het aantal benodigde manuren ten opzichte van het service window van de applicatie.
Meetinstrument	Uren registratie.
Doel	< 0,25
Wens	< 0,1

TABEL 7.

deze nog weinig richting voor de applicatie-ontwikkelaar. Hiervoor moet het begrip beheerbaarheid voorzien worden van eisen voor bijvoorbeeld het elimineren van handmatige procedures, de kwaliteit van de logmeldingen, het runtime kunnen aanpassen van parameters voor bijvoorbeeld logging of het runtime kunnen flushen van caches. Alle gedefinieerde eisen moeten een aantoonbare bijdrage leveren aan het bereiken van het doel op het hogere niveau, bijvoorbeeld:

Eis	Onderhoudbaarheid.beheerbaarheid.loglevel
Essentie	Het snel kunnen aanpassen van het loglevel binnen een class of package ten behoeve van probleemanalyse.
Schaal	Minuten service interruptie als gevolg van de aanpassing.
Meetinstrument	Daadwerkelijke test.
Doel	0 minuten: runtime aanpassing, zonder herstart.

TABEL 8.

Herbruikbaarheid. Herbruikbaarheid is voor beheerpartijen geen directe eis. Toch levert een herbruikbare

ADVERTENTIE

component stabielere software op doordat de software meer wordt gebruikt en dus impliciet beter getest is. Er kunnen heel goed meetinstrumenten worden gedefinieerd voor herbruikbaarheid. Denk bijvoorbeeld aan het aantal applicaties dat een specifieke softwarecomponent gebruikt.

PORTABILITEIT Portabiliteit is de overdraagbaarheid van de applicatie van de ene omgeving of technologie naar de andere.

Aanpasbaarheid. Dit betreft de mate waarin de applicatie overgezet kan worden naar andere technische omgevingen. Voor J2EE-applicaties wordt hier al in belangrijke mate in voorzien door de J2EE-specificatie. Vaak worden daarom geen specifieke eisen gesteld aan aanpasbaarheid. Veel applicatieservers implementeren echter extensies op de J2EE-specificatie die niet zonder meer uitwisselbaar zijn. Vanuit deze eigenschap bekeken is het raadzaam het gebruik van deze extensies zoveel mogelijk te beperken. Daarnaast is het bij een overgang van een oudere naar een nieuwe versie van een applicatieserver vaak zo dat oude constructies niet meer werken (denk aan JNDI lookups). Dit is een direct gevolg van het feit dat de nieuwe versie ook een nieuwe versie van de J2EE-standaard zal implementeren.

Installeerbaarheid. De deployment van J2EE-applicaties levert vaak problemen op: er moeten veel handmatige stappen worden uitgevoerd, een applicatie die in de ontwikkelomgeving goed werkt geeft onverwachte problemen in de acceptatieomgeving, en er is onvoldoende inzicht in welke versie van een applicatie in welke OTAP-omgeving staat geïnstalleerd. Veel organisaties zouden baat hebben bij een gestandaardiseerde, geautomatiseerde build en deployment. Door middel van bijvoorbeeld Ant- of Maven-scripts en het gebruik van de door de applicatieserver beschikbaar gestelde scripting-omgeving kan dit gerealiseerd worden. Het formuleren van beheerisen op dit gebied ligt dan ook voor de hand.

Eis	Portabiliteit.installeerbaarheid.deployments.foutgevoeligheid
Essentie	Een applicatie moet foutloos op een willekeurige OTAP omgeving kunnen worden geïnstalleerd.
Schaal	% mislukte deployments.
Meetinstrument	Daadwerkelijke deployments.
Doel	< 5%
Wens	< 1%

TABEL 9.

Conformiteit. De mate waarin de applicatie voldoet aan algemene J2EE en organisatiespecifieke standaards en specificaties voor zover deze van invloed zijn op portabiliteit.

BETROUWBAARHEID Betrouwbaarheid wordt gedefinieerd als een verzameling eigenschappen die weer geeft in hoeverre de applicatie in staat is het gewenste serviceniveau te leveren gedurende een bepaalde periode. Veel attributen van betrouwbaarheid hebben een directe impact op de beheerbaarheid van de applicatie.

Bedrijfszekerheid. Vanuit de systems-engineering discipline wordt bedrijfszekerheid gedefinieerd als Mean Time Between Failure (MTBF). Deze MTBF leidt vervolgens weer tot allerlei sub-eisen op het gebied van bijvoorbeeld de testkwaliteit en -coverage, bijvoorbeeld:

Eis	Betrouwbaarheid.bedrijfszekerheid.testcoverage
Essentie	De dekkingsgraad van de functionele testen.
Schaal	% line coverage.
Meetinstrument	JCoverage rapportage over de functionele testen.
Doel	> 90%
Wens	100%

TABEL 10.

Foutbestendigheid. De foutbestendigheid wordt voor een belangrijk deel geleverd door de J2EE-infrastructuur. Wel is het belangrijk dat de applicatie hiermee om kan gaan. Het gebruik van clusteringfaciliteit van de applicatieserver stelt bijvoorbeeld eisen aan het opslaan van informatie in de servletsessie. Ook transactiemangement vergt een gedegen ontwerp zodat er geen data verloren kunnen gaan. Denk bijvoorbeeld aan het niet adequaat opvangen van concurrency-issues als gevolg van optimistic locking.

Herstelbaarheid. De belangrijkste indicator voor herstelbaarheid is de Mean Time To Repair (MTTR). Factoren die hierbij een rol spelen zijn onder andere opgesomd bij analyseerbaarheid, wijzigbaarheid en testbaarheid. Ook opstarttijden van de applicatieserver en de applicatie spelen een rol.

Beschikbaarheid. Beschikbaarheid is een afgeleide van bedrijfszekerheid en herstelbaarheid en kan gedefinieerd worden als: $MTBF / (MTBF + MTTR)$. De gewenste beschikbaarheid wordt normaliter vastgelegd in een Service Level Agreement en uitgedrukt in een percentage over het online service window van een applicatie.

Afhankelijk van het type applicatie varieert dit percentage meestal tussen 95 en 99,9 procent.

CONCLUSIE Het Extended ISO 9126 model voor software-productkwaliteit biedt een goede kapstok om intake-eisen vanuit beheersperspectief voor J2EE-applicaties te structureren. In dit artikel is een aanpak gegeven om deze beheereisen op een ondubbelzinnige manier vast te leggen. De lezer zal voor de eigen organisatie de lijst met beheereisen verder moeten detailleren en completeren. Met het opstellen van een volledige lijst beheereisen is men er echter nog niet. Projecten zijn gericht op functionaliteit en zullen zelden de tijd en budget hebben om invulling te geven aan alle beheereisen en wensen als zij hier niet bij worden geholpen. Zonder verdere ondersteuning van beheer wordt een lijst beheereisen al snel een papieren tijger: wel de lasten van de extra inspanning, maar zonder de voordelen waar het allemaal om te doen was.

Ervaring leert dat het verstandig is de J2EE-beheereisen zoveel mogelijk te vertalen naar standaardoplossingen die door de diverse projecten gebruikt kunnen worden. De beheerafdelingen moeten hierbij het initia-

tief nemen en deze oplossingen (laten) ontwikkelen. Daarbij kan gedacht worden aan:

- Een standaard OTAP-omgeving met gewenste robuustheid en beschikbaarheid.
- Standaardscripts voor automatische deployment over de verschillende OTAP-omgevingen.
- Configuratiemanagement voor J2EE-omgevingen.
- Logging richtlijnen en utility's voor bijvoorbeeld het dynamisch aanpassen van log-levels.

Referenties

- [1] Zeist, B van e.a., *Kwaliteit van software producten - Ervaringen met een kwaliteitsmodel*, ISBN 92026724306.
- [2] Gilb, T., *Competitive Engineering – A handbook for systems engineering, requirements engineering, and software engineering using Planguage*, ISBN 0750665076.

Richard Swart is werkzaam bij Xebia.