

5.5 Analytic Trouble Shooting® The best practice for problem solving

Voor het oplossen van (al dan niet technische) problemen, bestaat al ruim 40 jaar een best practice. Het betreft een manier van problemen oplossen die bekend is als de Kepner-Tregoe methode (Kepner & Tregoe, 1997), *ProbleemAnalyse of Analytic Trouble Shooting® (ATS)*. De laatste jaren is deze methode zich in IT-omgevingen aan het ontwikkelen als een de facto standaard.

In dit artikel wordt de ATS-methode uitvoerig beschreven, alsmede de wijze van implementeren en de beoogde resultaten. Eerst wordt echter beschreven wat de relevantie is van rationele denkvaardigheden in organisaties.

Auteurs: Ing. Jan Willem Vernhout, Consultant/trainer Kepner-Tregoe en Ir. Ron Vonk, Managing consultant/trainer Kepner-Tregoe

HET TOENEMENDE BELANG VAN GESTRUCTUREERDE METHODEN VOOR TROUBLESHOOTING

Uitdagingen in CS-organisaties

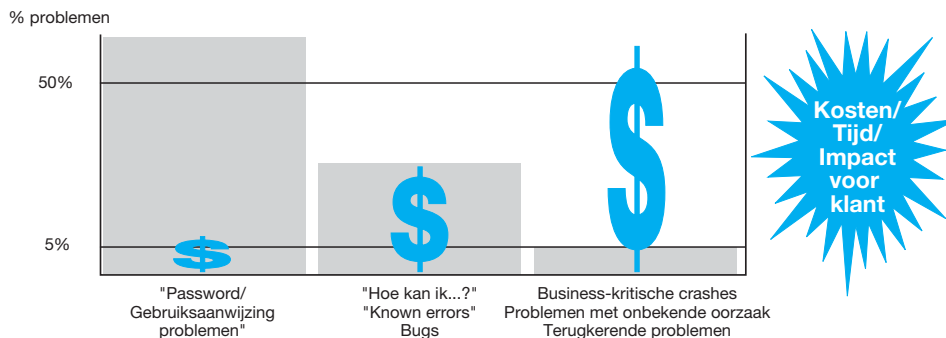
Alle fraaie procedures voor Incident- en Problemmanagement ten spijt, er zullen uiteindelijk nog steeds oorzaken van klantproblemen moeten worden gevonden. Goede procedures zorgen er immers vooral voor dat de juiste prioriteit wordt toegekend aan het probleem, de juiste routing voor het probleem wordt gekozen, de juiste mensen zich met het oplossen van het probleem bezighouden, het probleem tot het juiste managementniveau wordt geëscaleerd, communicatie naar alle stakeholders adequaat verloopt, etc.

Bij het oplossen zelf wordt tegenwoordig, zeker bij IT-problemen, wel steeds meer en met succes gebruik gemaakt van slimme uitvraagscenario's, scripting systemen, "known-error databases", etc.

Helaas is de bijdrage van dit soort hulpmiddelen eindig. Als het immers om een pro-

bleem gaat dat zich nog niet eerder heeft voorgedaan of dat niet eerder als potentieel probleem is voorzien, dan leiden de genoemde hulpmiddelen niet tot de meest waarschijnlijke oorzaak en zal men veelal moeten terugvallen op het gebruik van kennis en ervaring.

Gelukkig ligt het percentage van deze echt lastige problemen ("business critical crashes", problemen met echt onbekende oorzaak ofwel steeds terugkerende problemen) doorgaans onder de 5%. Minder gelukkig is het dat juist deze 5% problemen verreweg de grootste gevolgen hebben in termen van klantimpact, managementaandacht en geld (zie figuur 1). En juist dan blijken de geïnstalleerde hulpmiddelen dus tekort te schieten..... Deze groep van problemen wordt dan ook wel aangeduid als de 5% bom.



Figuur 1 De 5% bom

Naast deze 5% bom (en deels als gevolg hiervan) zijn er een aantal min of meer 'klassieke' kenmerken te noemen die men bij vele Customer Service organisaties kan herkennen:

- er is grote tijdsdruk bij het oplossen van problemen, het duurt al snel te lang;
- grote incidenten worden slecht en inconsistent gemanaged (iedere manager begint zich bijvoorbeeld met het probleem te bemoeien);
- er vindt slechte (namelijk erg incomplete) interne communicatie plaats, met name bij overdracht van problemen van de eene naar een volgende lijn;
- er vindt slechte communicatie met de klant plaats (niet alle relevante vragen worden direct gesteld, men kan de klant niet duidelijk maken waar men in het oplossingsproces staat en dus ook geen zekerheid bieden over hoelang een en ander nog zal duren, etc.);
- de backlog van problemen is groot en groeiende, mede omdat problemen terugkomen en niet definitief worden opgelost;
- de klant- en de medewerkerstevredenheid nemen af.

Om bovenstaande uitdagingen het hoofd te bieden is in toenemende mate behoefte aan een gestructureerde, effectieve, consistente, praktische wijze van oplossen van (klant)problemen. Probleem is dat in de praktijk veelal alleen gebruik wordt gemaakt van onbewus-

te ofwel intuïtieve methoden van troubleshooting.

Intuïtieve versus gestructureerde methoden van troubleshooting

Bij het oplossen van problemen kan onderscheid worden gemaakt in een tweetal componenten die beiden noodzakelijk zijn om tot een oorzaak te komen. In figuur 2 is dit gevisualiseerd. Ten eerste zal men de nodige inhoudelijke kennis en ervaring en desgewenst aanvullende informatie over het systeem of de dienst waarmee een probleem is (het 'wat') moeten hebben. Daarnaast is echter een methode, systeem of logica nodig om de juiste (denkproces)stappen in de juiste volgorde te zetten (het 'hoe').

Inhoudelijke kennis en ervaring is veelal voldoende aanwezig of weet men in ieder geval te vinden. Daarvoor worden medewerkers ook continu opgeleid en juist daarvoor worden ondersteunende softwareapplicaties geïntroduceerd. Wat doorgaans veel beperkter aanwezig is, is een expliciete (denk) methode voor het oplossen van het probleem.

Troubleshooters hebben vaak een eigen, veelal onbewuste of intuïtieve, methode voor het oplossen van problemen. Zeer herkenbaar is het dat men bij het hanteren van deze eigen methode vaak een 'jump to conclusions' maakt op basis van inhoudelijke ken-

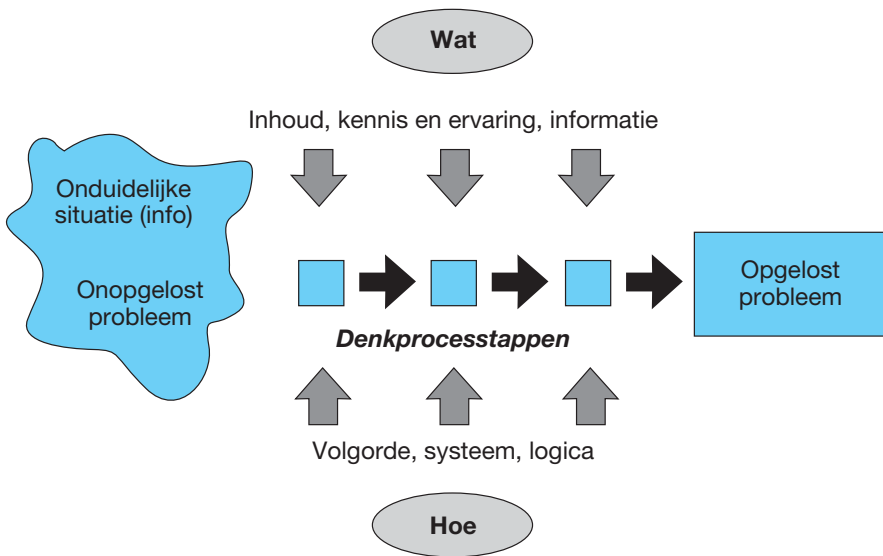
Intuïtieve methoden	Gestructureerde methoden
<ul style="list-style-type: none"> • Ongestructureerd, anders voor iedere troubleshooter • Onbewust eigen gemaakt en daardoor lastig te verbeteren • Verspillen tijd, geld en klanttevredenheid indien toegepast op alle problemen • Zijn 'onzichtbaar' en daardoor lastig te managen • Leiden doorgaans tot meerdere (soms vele) achtereenvolgende reparatiepogingen • Zijn veelal gericht op symptomen, waardoor de oorzaken kunnen terugkomen • Kunnen, zeker bij herhaalde pogingen, tot irritatie en weerstand tegen een volgende oplossing leiden 	<ul style="list-style-type: none"> • Gestructureerd: Komen van geobserveerd 'best practices' in denken • Bewust aan te leren methode (onbewuste vaardigheid komt pas bij veel praktiseren) • Verschaffen een alternatieve aanpak bij écht lastige problemen • Maken voortgang in het troubleshoot-proces zichtbaar en daardoor incident- en probleemmanagement veel eenvoudiger • Bewijzen eerst de oorzaak alvorens een oplossing aan te dragen • Verwijderen de oorzaak van terugkerende problemen • Vergroten het vertrouwen van de klant in de oplossing die wordt voorgesteld

Tabel 1 Kenmerken intuïtieve methoden en gestructureerde methoden voor Troubleshooting

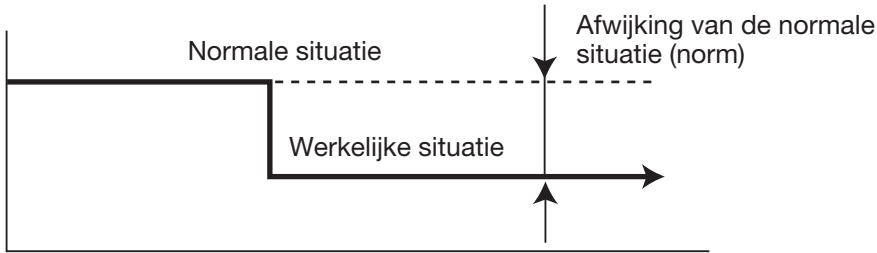
nis en ervaring. Maar er zijn meer nadelen te noemen. In tabel 1 zijn de verschillen tussen intuïtieve en gestructureerde wijzen van troubleshooting weergegeven.

Is het gebruik van kennis en ervaring, en zelfs 'jumping to conclusions' altijd verkeerd? Zeker niet. 'Jumping to conclusions' is en

blijft de meest efficiënte (snelle) methode om tot de oorzaak van een probleem te komen... als men gelijk blijkt te hebben. Probleem is dat men vaak niet herkent dat men in een blijvend proces van 'jumping to conclusions' terecht is gekomen met de ene na de andere (dure) reparatiepoging als gevolg. Er is in die situaties behoefte aan een *trigger* die aan-



Figuur 2 Inhoud versus proces



Figuur 3 Afwijking van de norm

geeft dat een alternatieve methode noodzakelijk is om verder tijdsverlies, kosten en klantirritatie te voorkomen. Vervolgens is uiteraard vaardigheid in die alternatieve methode noodzakelijk.

In het volgende onderdeel wordt de Analytic Trouble Shooting methode beschreven als dé alternatieve methode voor het oplossen van problemen. Over de genoemde 'trigger' wordt meer geschreven in het onderdeel 'Implementatie ATS in organisaties', waar de verankering van de methode in de dagelijkse praktijk aan de orde komt.

ANALYTIC TROUBLE SHOOTING®

Hoewel Analytic Trouble Shooting® formeel meerdere stappen omhelst (Visser & Vonk, 2001), wordt er in deze paragraaf alleen aandacht geschonken aan één denkproces: ProbleemAnalyse, het denkproces dat gericht is op het vinden van een oorzaak van een probleem. In deze paragraaf wordt dit denkproces uitgebreid behandeld. Eerst wordt echter een definitie van een probleem gegeven, een probleem waarvoor het zinvol is ProbleemAnalyse toe te passen. Aan het einde van dit artikel is een volledig uitgewerkt voorbeeld weergegeven.

NB: De methode is universeel toepasbaar. Het maakt daarom niet uit of deze wordt toegepast op een systeem, computer, installatie, product, dienst, mens, of wat dan ook. Gemakshalve zullen wij in het vervolg steeds

spreken van een systeem als 'het iets' waarmee een probleem is. Verder vermelden wij steeds 'troubleshooter' waar sprake kan zijn van een customer-support medewerker, engineer, technician, etc.

Definitie van een probleem

De taak van een troubleshooter is het vinden van de oorzaken van incidenten of problemen. Maar voordat de oorzaak gevonden kan worden moet er duidelijkheid zijn over het probleem. Om te weten of er daadwerkelijk sprake is van een probleem moet bekend zijn hoe het systeem behoort te werken. Als de werkelijke prestaties van een systeem niet overeenkomen met de gewenste prestaties en we weten niet waarom, dan is er sprake van een probleem (zie figuur 3).

Wetende wat er had moeten gebeuren, de normale situatie, kan een probleem worden herkend door deze situatie te vergelijken met dat wat er nu gebeurt, de werkelijke situatie. Als er een verschil is tussen de normale situatie en de werkelijke situatie, dan is er een probleem, ofwel een afwijking van de norm. Hoe scherper de omschrijving van wat er had moeten gebeuren, hoe duidelijker de herkenning van een afwijking als die zich voordoet.

Het hebben van een afwijking ten opzichte van de norm met een onbekende oorzaak wil nog niet altijd zeggen dat het zinvol is om ProbleemAnalyse toe te gaan passen. Het moet ook noodzakelijk zijn de werkelijke oorzaak te kennen om een maatregel te kunnen bepalen die het probleem wegneemt. Dat

hoeft (nog) niet altijd het geval te zijn. Denk bijvoorbeeld aan het resetten van een systeem om een symptoom weg te nemen. Als het betreffende probleem zich echter blijft herhalen is het wenselijk de echte oorzaak te kennen, zodat een duurzame maatregel kan worden genomen. In dat geval is het gebruik van ProbleemAnalyse dus wel noodzakelijk.

ProbleemAnalyse

Indien er een verschil is tussen wat gebeurt en hetgeen wat had moeten gebeuren, het is onbekend waarom dit precies optreedt en de oorzaak moet gevonden worden dan is er sprake van een probleem. ProbleemAnalyse is nu de aangewezen manier om de juiste oorzaak te achterhalen, maar eveneens om te voorkomen dat tijd en energie gestopt wordt in het oplossen van 'schijnbare' oorzaken.

Het proces van ProbleemAnalyse kent de volgende logische stappen:

1. Beschrijf het probleem;
 - a. geef probleemomschrijving
 - b. specificeer het probleem.
2. Formuleer mogelijke oorzaken;
 - a. op basis van kennis en ervaring
 - b. op basis van kenmerken en veranderingen.
3. Evalueer mogelijke oorzaken;
 - a. toets mogelijke oorzaken
 - b. bepaal meest waarschijnlijke oorzaken.
4. Bewijs de juiste oorzaak.

Stap1: Het beschrijven van het probleem

Stap 1a: Geef Probleemomschrijving

Een goede beschrijving van het probleem is absoluut noodzakelijk. Voordat het probleem opgelost kan worden moet bekend zijn wat precies het probleem is. Als al met oplossen wordt begonnen als men enkel maar een algemeen idee heeft over wat het probleem is, wordt de kans een stuk kleiner dat het probleem wordt opgelost. Om een begin van de beschrijving te maken moeten de volgende vragen worden gesteld:

“Welk systeem of welke systemen hebben een probleem?”

“Welke afwijking hebben ze?”

Dit resulteert in een korte beschrijving van het probleem om het probleem centraal te stellen. Een probleemomschrijving wordt altijd genoteerd in het 'format' van een object en een afwijking.

Het object is het systeem (of groep van systemen) waar problemen mee zijn. De afwijking is wat er mis mee is. Om op het goede spoor te blijven dienen zowel het object als de afwijking zo specifiek mogelijk in de probleemspecificatie beschreven te worden.

Voorbeelden:

- Klant x kan geen verbinding krijgen.
- Server 7 wordt steeds trager.
- Systeem C is vastgelopen.

Stap 1b: Specificeer het probleem

Om een probleem te specificeren moet er door het stellen van een reeks vragen informatie worden verzameld op een viertal gebieden: WAT (identiteit), WAAR (locatie), WANNEER (tijdstip) en OMVANG (impact). De hierdoor verkregen beschrijving van het probleem noemen we de probleemspecificatie. Uniek aan de ATS-methode is het specificeren van het probleem in termen van IS en IS-NIET. De vragen op elk van de vier gebieden over wat het probleem wél IS worden hiertoe vergeleken met vragen over wat het probleem zou kunnen zijn, maar wat het NIET IS. De antwoorden die worden gezocht bij het beantwoorden van IS-NIET vragen liggen bij voorkeur zo dicht mogelijk tegen de IS-antwoorden aan, zijn zoveel mogelijk daaraan verwant. Wanneer er antwoord is op alle IS en IS NIET vragen volgt een accurate en complete probleemspecificatie.

In tabel 2 volgt een complete lijst van vragen over wat het probleem IS en wat het probleem NIET IS.

	Is	Is niet
Wat	<ul style="list-style-type: none"> • Welk object, groep van objecten, ding, product, etc. heeft een probleem? • Wat is er mis mee? Wat is de afwijking? 	<ul style="list-style-type: none"> • Welk object, groep van objecten, ding, product, etc. had dit probleem ook kunnen hebben, maar heeft het nu niet? • Welke andere afwijkingen hadden kunnen worden waargenomen maar blijken nu niet op te treden?
Waar	<ul style="list-style-type: none"> • Waar / op welke plaats was het object toen het probleem werd gezien? • Waar op het object bevindt zich de afwijking? 	<ul style="list-style-type: none"> • Waar / op welke plaats had het object met de afwijking kunnen zijn, maar is het niet? • Waar op het object had de afwijking zich kunnen bevinden, maar bevindt het zich niet?
Wanneer	<ul style="list-style-type: none"> • Wanneer werd het probleem voor het eerst opgemerkt (datum/tijd)? • Wanneer werd het probleem sindsdien waargenomen (datum / tijd / patroon)? • Wanneer in de levensloop of het proces werd het probleem voor het eerst opgemerkt? 	<ul style="list-style-type: none"> • Wanneer had het probleem ook voor het eerst opgemerkt kunnen worden, maar werd het niet geconstateerd? • Wanneer had het probleem sindsdien waargenomen kunnen worden, maar werd het niet geconstateerd? • Wanneer in de levensloop of het proces had het probleem opgemerkt kunnen worden, maar werd het niet geconstateerd?
Omvang	<ul style="list-style-type: none"> • Hoeveel objecten vertonen de afwijking? • Hoe erg of hoe groot is één afwijking? • Hoeveel afwijkingen zitten er op één object? • Wat is de trend? Meer, minder of gelijkblijvend aantal objecten of afwijkingen? 	<ul style="list-style-type: none"> • Hoeveel objecten hadden de afwijking kunnen vertonen, maar doen dat niet? • Hoe erg had één afwijking kunnen zijn, maar is dat niet? • Hoeveel afwijkingen hadden er op één object kunnen zitten, maar zijn er niet? • Wat had de trend kunnen zijn, maar is het niet?

Tabel 2 Specificatievragen voor probleembeschrijving

Niet alle vragen zullen altijd van toepassing lijken, maar het is belangrijk die vragen toch te stellen en er ook zoveel als mogelijk een antwoord op te krijgen door door te blijven vragen.

Specificeer het probleem - vraag WAT

- Welk object of groep objecten, ding, product, etc. heeft een probleem?
- Wat is er mis mee? Wat is de afwijking?

Er is al een antwoord op deze twee vragen gegeven in de probleemomschrijving. De vragen worden nogmaals gesteld om de spe-

cificatie - zoals het woord al zegt - wellicht specifieker te krijgen dan de probleemstelling. Vraag net zolang door tot de onderdelen waar problemen mee waren zo specifiek mogelijk omschreven zijn. Deze methode, waarbij dezelfde vraag net zo lang wordt gesteld tot alle beschikbare informatie is verkregen, heet *doorvragen tot op de kern*, een methode die toegepast dient te worden bij het stellen van alle specificatievragen.

Voorbeeld:

Klant: Mijn computer functioneert niet
Helpdesk: Wat functioneert er precies niet

aan uw computer?

Klant: Als ik applicatie X probeer op te starten gebeurt er niets

Helpdesk: Dus als ik het goed begrijp dan kunt u applicatie X niet opstarten?

Klant: Ja dat klopt

Vaak worden belangrijke feiten van het probleem over het hoofd gezien. Bij 'doorvragen' komt informatie over de afwijking naar boven die in eerste instantie aan de aandacht was ontsnapt. Aanwijzingen over de oorzaak van een probleem zijn vaak te vinden in het specifieke van de afwijking. Deze aanwijzingen blijven vaak onopgemerkt als de vragen niet gedetailleerd genoeg worden gesteld.

Lokaliseer het probleem - vraag WAAR

a. Waar / op welke plaats was het object toen het probleem werd gezien?

Met deze vraag wordt meer informatie gekregen over de geografische locatie. Op welke plaatsen of in welke ruimtes doet het probleem zich voor? Problemen doen zich vaak op meerdere plaatsen voor, en alle plaatsen waar die problemen zich voordoen dienen genoteerd te worden.

b. Waar op het object bevindt zich de storing?

Met deze vraag wordt het object bestudeerd om de exacte locatie van het probleem, de afwijking, of de storing vast te stellen. Maar soms is het antwoord op deze vraag moeilijker en zijn er geen voelbare of zichtbare afwijkingen, en kan er geen gebied worden afgebakend. Wanneer een server niet meer wil starten, wáár in de server bevindt zich dan het probleem? Of, als een netwerkkaart niet goed is, wáár in de netwerkkaart zit het probleem? In zulke gevallen heeft het weinig zin om deze vraag te stellen.

Stel de tijd vast van het probleem - vraag WANNEER

Wanneer werd het probleem voor het eerst opgemerkt (datum, tijd)?

Nu dient de datum en het tijdstip waarop het probleem voor het eerst door iemand werd opgemerkt achterhaald te worden. Zoals met

alle andere antwoorden moet ook hier de informatie zo specifiek mogelijk zijn. Noteer niet "vanmorgen" als bij doorvragen bekend kan worden hoe laat het precies was. Wees niet tevreden met "ergens tussen negen en tien uur" maar vraag of iemand misschien weet wanneer precies tussen negen en tien uur. Een typische valkuil bij een Helpdesk: men noteert alleen de tijd van melding van het probleem door de klant. In dit geval zoeken we echter naar het tijdstip waarop de klant *zelf* het probleem voor het eerst waarnam.

a. Wanneer werd het probleem sindsdien waargenomen (datum, tijd, patroon)?

Na het vaststellen van het tijdstip waarop de afwijking zich voor het eerst voordeed, moet het ook bekend worden of het probleem zich al vaker heeft voorgedaan. Ook nu moeten de datum en het tijdstip waarop het probleem zich iedere keer heeft voorgedaan zo compleet en zo exact mogelijk achterhaald worden. Uit deze feiten kan een duidelijk omljnd patroon te voorschijn komen.

Als het niet mogelijk blijkt de exacte data en tijdstippen te achterhalen, gebruik dan de best beschikbare informatie. "Sinds afgelopen januari is dit twee of drie keer voorgekomen." "Het komt af en toe voor." Bij voorkeur zou de informatie specifiekere moeten zijn, maar voorlopig valt er niet meer over te vertellen. Noteer dit, maar vergeet niet dat de informatie vaag is en blijf doorzoeken naar betere informatie.

Probeer antwoorden te krijgen op de vraag "Wanneer... voor het eerst" en "Wanneer... sindsdien" in tijdsaanduiding en datum. "Woensdag de 12e om 9.15 uur" is het soort antwoord dat we op de eerste twee vragen willen. Het doel van deze twee vragen is om het meest accurate tijdstip te krijgen waarop het probleem zich heeft voorgedaan.

De volgende WANNEER-vraag is bedoeld om op een andere manier het tijdstip van het probleem te bepalen. Het levert een tijdstip op dat met andere gerelateerde gebeurtenissen

in verband kan worden gebracht. Samenhangende gebeurtenissen zijn erg belangrijk, en daarom moet de vraag over de levensloop van het voorwerp worden gesteld.

b. Wanneer in de levensloop of het proces werd het probleem voor het eerst opgemerkt?

Alle dingen of systemen hebben een geschiedenis, een 'levensloop'. De levensloop is een verslag van alle gebeurtenissen van het systeem gerekend vanaf moment dat het als idee of ontwerp is ontstaan tot aan het moment dat het systeem ophield te bestaan. Door de vraag over de levenscyclus van het systeem te stellen is er informatie over andere gebeurtenissen die zich hebben voorgedaan net voordat of op het moment dat het probleem zich presenteerde. Deze gebeurtenissen maken deel uit van de geschiedenis van het systeem dat het probleem heeft.

Er kunnen zich op elk moment in de levenscyclus problemen voordoen. Het helpt als bekend is wanneer in de levenscyclus het probleem *voor het eerst* werd opgemerkt.

Voorbeelden:

Tijdens opstarten van de server
Na de change, uitgevoerd door afdeling ...
Direct na het intoetsen van ...

Luister aandachtig naar de antwoorden op de WANNEER-vragen, en controleer of ze werkelijk WANNEER-antwoorden zijn of alleen maar gissingen naar de oorzaak.

Taxeer het probleem - vraag naar de OMVANG

a. Hoeveel objecten vertonen de afwijking?

Het stellen van deze vraag is alleen zinnig als het gaat om een groep gerelateerde voorwerpen.

b. Hoe erg of hoe groot is één afwijking?

Ook dit is een vraag die soms wel en soms niet relevant is. De vraag "Hoe erg hangt de computer?" is niet zo zinvol. De vraag "Hoe groot is de verminderde transmissiecapaciteit?" echter weer wel.

c. Hoeveel afwijkingen zitten er op één object?

Deze vraag kan alleen worden beantwoord wanneer het gaat over een meervoudige afwijking.

d. Wat is de trend? Meer, minder of een gelijkblijvend aantal objecten of afwijkingen?

Met deze vraag wordt onderzocht of het probleem erger wordt, minder erg wordt of hetzelfde blijft. Als er specifieke data en tijdstippen beschikbaar zijn waarop het probleem zich heeft voorgedaan, en als ook bekend is hoe erg het elke keer was, kan de voortgang worden uitgetekend of uitgerekend. Als de voortgang van een probleem bekend is, dan kan daarmee de oorzaak gemakkelijker gevonden worden. Bestudeer de ontwikkeling in het systeem. Als er ieder uur meer klantenklachten binnenkomen, dan moet de mogelijke oorzaak deze vermeerdering kunnen verklaren. Kijk dan ook naar de voortgang of groei in de afwijking. Als de klachten steeds ernstiger worden, dan ligt het aan de mogelijke oorzaak.

Enkele opmerkingen over IS NIET

Waarom zoveel moeite doen om te achterhalen wat het probleem NIET IS? Wat is de toegevoegde waarde van deze IS NIET informatie? Als bekend is wat het probleem NIET IS zijn er twee manieren bijgekomen om de oorzaak van het probleem te bepalen. Met de IS NIET vragen is er een beter inzicht in de antwoorden op de IS vragen. Door de IS NIET vragen te beantwoorden komt het accent te liggen op de informatie over wat het probleem WEL IS, het verscherpt de IS informatie.

Daarnaast bakent de IS NIET informatie het probleem af en beperkt de mogelijke oorzaken. Hoe dichter de IS en de IS NIET bij elkaar liggen, des te duidelijker is het probleem afgebakend.

Voorbeeld:

Stel, u kijkt naar de televisie en kanaal 1 waarnaar u kijkt valt in ruis. Wat is het eerste wat u doet? Juist, zappen naar kanaal 2. En als

dat ook in ruis is? Verder zappen? Overal ruis? Kijken naar de televisie op de slaapkamer.

Wat u doet is het zoeken naar IS-NIET'en die zo dicht mogelijk liggen tegen de IS, omdat dat u het meeste helpt bij het vinden van de oorzaak.

Stap 2: Mogelijke oorzaken bepalen

Als er eenmaal een goede beschrijving van het probleem is wordt een lijst gemaakt van de mogelijke oorzaken van het probleem. Daarvoor zijn twee methoden voorhanden. Men kan simpelweg antwoord proberen te krijgen op de vraag: "Wat kan de oorzaak zijn van dit probleem?". Soms blijken er echter geen oorzaken (meer) te bedenken of zijn er juist heel veel mogelijke oorzaken te formuleren. In beide gevallen kan gebruik worden gemaakt van een tweede methode en wel het gebruik van Kenmerken en Veranderingen. In deze paragraaf worden deze twee methoden beschreven.

Stap 2a: Gebruik kennis en ervaring om mogelijke oorzaak te formuleren

Nadat de probleemomschrijving en de probleemspecificatie zijn genoteerd moet de kennis van het systeem gebruikt worden om een lijst te maken van de *mogelijke oorzaken* van het probleem. Vraag: "Wat kan de oorzaak zijn van dit probleem?" Schrijf daarna de mogelijke oorzaken op in het format van 'object en afwijking', zoals dat ook reeds bij de probleemomschrijving werd gedaan, aangevuld met een korte/globale verklaring hoe deze oorzaak heeft kunnen leiden tot het probleem.

Veelal is de probleemomschrijving al zo duidelijk en de kennis en ervaring zodanig voorhanden dat direct mogelijke oorzaken kunnen worden bedacht. Wat eveneens helpt is het bekijken van de probleemspecificatie.

Stap 2b: Gebruik kenmerken en veranderingen om mogelijke oorzaak te formuleren

De mogelijke oorzaken van een probleem kunnen ook bedacht worden aan de hand

van de feiten in de probleemspecificatie. Het belangrijkste hier is om de kenmerken en de veranderingen vast te stellen. Zoek het verschil tussen de IS en de IS NIET informatie uit uw probleemspecificatie, en zoek dan naar de veranderingen die te maken hebben met deze kenmerken. Een scherp en duidelijk verschil kan leiden naar een verandering en uiteindelijk naar een meest waarschijnlijke oorzaak.

Zoek de kenmerken

Na het specificeren van een probleem kan uit eigen ervaring misschien al bedacht worden wat de mogelijke oorzaak zou kunnen zijn. Die kan dan meteen worden getoetst. Als één van de mogelijke oorzaken erg waarschijnlijk lijkt, verifieer die dan in de praktijk. Maar als geen van de mogelijke oorzaken de toets doorstaat, ga dan terug naar de probleemspecificatie en zoek de kenmerken. Wanneer er een probleem is gespecificeerd en er is geen enkele mogelijke oorzaak te bedenken, ga dan meteen door met het zoeken naar de kenmerken.

Het kan zijn dat er een groot aantal mogelijke oorzaken is bedacht, maar dat geen van die mogelijke oorzaken de werkelijke oorzaak is; ook kan het gebeuren dat er überhaupt geen enkele mogelijke oorzaak bedacht kan worden. Dit komt vaak voor als een probleem te maken heeft met een nieuw systeem of met een systeem waar niets over bekend is.

Om mogelijke oorzaken te vinden dient gekeken te worden naar de IS en IS-NIET informatie uit de probleemspecificatie. Overweeg, en vraag ook aan anderen: "Wat is het kenmerk tussen het IS feit ten opzichte van het IS NIET feit?"

Als deze vraag geen bevredigend antwoord oplevert, stel de vraag dan op een andere manier. In plaats van te vragen wat het kenmerk is, kan er bijvoorbeeld worden gevraagd naar wat er vreemd, ongewoon, eigenaardig, bijzonder, uniek of uitzonderlijk is voor IS in vergelijking met IS NIET.

Als de IS gegevens met de IS NIET gegevens worden vergeleken kan het zo zijn dat er meer dan één kenmerk naar boven komt. Noteer altijd alle gevonden kenmerken. Blijf wel doorzoeken naar nieuwe informatie, want er zijn nieuwe feiten nodig die nog niet in de probleemspecificatie staan. Tenslotte is de probleemspecificatie een opsomming van feiten, terwijl de kenmerken een analyse van die feiten weergeven. Het zou tijdsverspilling zijn als de feiten simpelweg worden herhaald.

Bij het zoeken naar kenmerken is alleen het kenmerk in de IS informatie van toepassing. Als het kenmerk zowel voorkomt bij de IS als bij de IS NIET, is het geen kenmerk meer.

Het vinden van gebieden van scherpe contrasten

Er is hierboven vermeld dat er gezocht moet worden naar de kenmerken in elk paar IS en IS-NIET informatie in de probleemspecificatie. Het beste is om de hele specificatie door te kijken. Als er geen kenmerk te vinden is, forceer het dan niet. Meestal zijn er geen (nuttige) kenmerken te zien in de AFWIJKING- en de OMVANG-informatie. Scherpe contrastgebieden zijn het gemakkelijkst waar te nemen als er een scherpe probleemspecificatie is gemaakt. Hoe dichter de IS-NIET bij de IS ligt, hoe waarschijnlijker het is dat er een bruikbaar kenmerk te zien is. Meestal is er in een specificatie tenminste één paar IS en IS-NIET informatie dat er werkelijk uitspringt. Dat is vaak het geval als de IS en de IS NIET erg op elkaar lijken. Zoek de

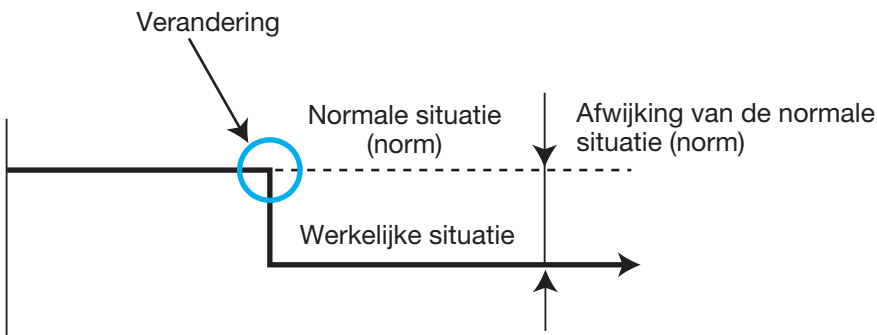
IS en IS-NIET-paren waar het kenmerk meteen opvalt en om uitleg vraagt. Dit zijn de gebieden van scherpe contrasten, en in elk van deze gebieden kan een belangrijk kenmerk schuilen dat naar de oorzaak kan leiden.

Zoeken naar veranderingen

Eerder is al gezegd dat een probleem een afwijking is tussen de NORM en de WERKELIJKHEID, en dat de oorzaak daarvan gevonden moet worden. Dit gaf de schematische voorstelling op zoals weergegeven in figuur 3, waarin is te zien dat er een periode was dat er geen afwijking bestond: de normale situatie (de norm) en de werkelijke situatie (de werkelijkheid) waren gelijk. Als er niets was veranderd, en de norm op het niveau gebleven was van de werkelijkheid, dan hadden we nu geen probleem gehad. Maar er is wél wat veranderd, en dat is toegevoegd aan het onderstaande schema in figuur 4: een pijl om aan te geven dat er een verandering heeft plaats gevonden.

Een *verandering* heeft plaatsgevonden wanneer iets op een bepaald moment anders is. Stel dat de oorzaak van de afwijking niet bekend is, maar wel bekend moet worden. Als we in het verleden konden kijken en te weten konden komen wat de verandering was, dan zouden we ook weten wat de oorzaak van het probleem was. In de meeste gevallen zijn veranderingen dus belangrijk voor het ontwikkelen van mogelijke oorzaken.

Het zoeken naar veranderingen kan veel tijd



Figuur 4 Verandering als oorzaak van een afwijking van de norm

in beslag nemen. Elke minuut veranderen er honderden dingen in onze omgeving. Als er zich zes maanden geleden een probleem heeft voorgedaan, probeer dan maar eens terug te gaan in de tijd en een lijst te maken van alle dingen die omstreeks die tijd zijn veranderd! Zelfs al zou iemand zich alles nog kunnen herinneren, dan kan het toch nog zo zijn dat 99% van die veranderingen niets met het probleem te maken had!

Dat is de reden waarom het beter is niet gelijk naar veranderingen te zoeken zodra er een probleemspecificatie is opgesteld. Eerst wordt er gezocht naar de kenmerken - unieke aspecten van het IS - en daarna pas naar veranderingen in die unieke aspecten. Dit beperkt het aantal veranderingen dat onderzocht moet worden en kan er met zekerheid gezegd worden dat de overblijvende veranderingen inderdaad met het probleem te maken kunnen hebben.

Als de lijst met kenmerken is opgesteld, bekijk die kenmerken dan stuk voor stuk en vraag: "Wat is er veranderd in, op of in de buurt van dit kenmerk?" De vraag kan ook anders worden gesteld: "Wat is er verbeterd, aangepast, veranderd, vooruitgegaan, gemoderniseerd, begonnen, gestopt, bijgesteld, toegevoegd, gemaakt of gecorrigeerd?" Als daar geen antwoord op komt, controleer dat dan en noteer achter het kenmerk dat bekeken is dat er 'geen veranderingen bekend' zijn.

Vraag ook altijd naar de datum en de tijd van de veranderingen die zich hebben voorgedaan (en noteer het!). Als de mogelijke oorzaken worden getoetst aan de probleemspecificatie is het belangrijk te weten wanneer die verandering heeft plaatsgevonden. Bijvoorbeeld, als de verandering heeft plaatsgevonden nadat het probleem begon, dan kan die verandering niet het probleem veroorzaakt hebben.

Gebruik de kenmerken en de veranderingen om mogelijke oorzaken uit te werken
Omdat het doel van het noteren van ken-

merken en veranderingen is om mogelijke oorzaken van het probleem te ontwikkelen, is het belangrijk de bevindingen om te zetten in mogelijke oorzaken. Noem bij de beschrijving van de mogelijke oorzaak ook een object en een afwijking, zoals al is gedaan bij de eerste probleemstelling. Let er op dat iedere probleemstelling ook verklaart hoe de oorzaak het probleem kan veroorzaken.

Om de mogelijke oorzaken aan te wijzen en te noteren, kunnen de volgende vragen worden gesteld:

- *Hoe kon deze verandering het probleem veroorzaken?*
- *Hoe kon deze verandering plus dit kenmerk het probleem veroorzaken?*
- *Hoe kon deze verandering plus die verandering het probleem veroorzaken?*
- *Hoe kon dit kenmerk het probleem veroorzaken?*

Stel de vragen in deze volgorde. De meeste oorzaken komen voort uit slechts één verandering, of uit een verandering plus een kenmerk. En stel de vragen eerst over iedere verandering, en dan over iedere verandering plus kenmerk samen.

Stap 3: Evalueren mogelijke oorzaken

Stap 3a: Toets de mogelijke oorzaken aan de specificatie

Een gedegen troubleshooter baseert mogelijke oorzaken op feiten. Jammer genoeg willen sommige troubleshooters nog wel eens een oorzaak aanwijzen voordat de feiten bekend zijn, en blijven dan ook nog volhouden dat ze gelijk hebben. "Ik weet al wat er aan de hand is: bij de onderhoudsbeurt is de kaart niet goed in het slot gestoken en daarom ontstaan er foute signalen," Een aardige gok, maar niet altijd juist. Door de Probleem-Analyse methode toe te passen wordt de oorzaak niet zomaar uit de lucht gegrepen. Door de oorzaken te toetsen aan de probleemspecificatie worden eventuele voorbarige conclusies geëlimineerd, en kan veel tijd, geld en moeite worden bespaard.

Het terugbrengen van de lijst van mogelijke oorzaken naar die ene en werkelijke oorzaak hangt af van de mogelijkheid of iedere oorzaak getoetst kan worden aan de IS / IS NIET informatie uit uw probleemspecificatie. Deze toetsing neemt slechts een paar minuten in beslag maar kan vele uren werk besparen. Nadat het probleem is gespecificeerd en een mogelijke oorzaak is bedacht moet nagegaan worden hoe deze oorzaak het probleem heeft kunnen veroorzaken.

De vraag die gesteld moet worden is: Als de oorzaak is van, hoe verklaart dit dan de IS en de IS NIET? U stelt deze vraag voor alle IS / IS NIET informatie die tot uw beschikking staat.

Bij het toetsen van een mogelijke oorzaak worden alle gegevens meegenomen die in de probleemspecificatie voorkomen. Per regel in de specificatie wordt er geverifieerd of deze mogelijke oorzaak de antwoorden op de IS / IS NIET vragen kan verklaren. Dat is of het geval, of niet het geval, of het is het geval als u uitgaat van een bepaalde veronderstelling. Als ergens in de specificatie blijkt dat deze waarschijnlijke oorzaak op geen enkele manier een verklaring geeft voor de feiten, dan is dat niet de oorzaak van het probleem. Als een oorzaak de gegevens alleen maar op grond van aanvullende veronderstellingen kan verklaren, moet die mogelijke oorzaak niet direct worden geëlimineerd, maar is de kans dat het de werkelijke oorzaak is wel kleiner. Hoe meer veronderstellingen er bedacht moeten worden en hoe vager deze zijn, hoe onwaarschijnlijker het wordt dat hiermee de ware oorzaak van het probleem wordt gevonden.

Stap 3b: Bepaal de meest waarschijnlijke oorzaak

Stel dat er vijf mogelijke oorzaken van een probleem zijn getoetst. Twee daarvan zijn afgevallen omdat die de feiten uit de probleemspecificatie niet konden verklaren. De andere drie konden ermee door omdat er enkele veronderstellingen werden gemaakt. Van die drie had de eerste mogelijke oorzaak

vijf aannames nodig, de tweede had er drie nodig, en de laatste slechts één aanname. In de meeste gevallen zou deze derde mogelijke oorzaak de *meest waarschijnlijke oorzaak* zijn. Deze oorzaak verklaart namelijk de IS / IS NIET feiten met het minst aantal aannames. Daarom is dit de mogelijke oorzaak die het eerst bewezen gaat worden. Natuurlijk is de meest waarschijnlijke oorzaak niet altijd te selecteren aan de hand van het aantal aannames. Soms is het zo dat een oorzaak met drie redelijke aannames aanvaardbaarder is dan een oorzaak met één enkele aanname. Gebruik uw beoordelingsvermogen om de meest waarschijnlijke oorzaak te selecteren. Vergeet niet dat het hier gaat om de keuze van de oorzaak die *het eerst* moet worden bewezen.

Wanneer de geselecteerde oorzaak niet de werkelijke oorzaak blijkt te zijn, kijk dan opnieuw naar de lijst en selecteer de daaropvolgende meest waarschijnlijke oorzaak. Het toetsen van de oorzaken aan de probleemspecificatie geeft nog geen garantie dat het antwoord meteen wordt gevonden. Het maakt het wel een stuk eenvoudiger en goedkoper dan wanneer bij elke mogelijke oorzaak meteen tot actie wordt overgegaan in de hoop dat er iets uit komt.

Stap 4: Bewijzen van de juiste oorzaak

Nadat de meest waarschijnlijke oorzaak is geselecteerd, is de volgende stap het verifiëren, ofwel bewijzen dat dit ook echt de oorzaak is. Het bewijzen van de werkelijke oorzaak is wat gedaan moet worden om aan te tonen dat de meest waarschijnlijke oorzaak de werkelijke oorzaak is. Het controleren van de juistheid van de werkelijke oorzaak is een belangrijke stap: wanneer de waarschijnlijke oorzaak niet bewezen wordt, zal het niet duidelijk worden of dat ook echt de werkelijke oorzaak is. En als niet met zekerheid gezegd kan worden wat de werkelijke oorzaak is, kan het zo zijn dat er een maatregel wordt genomen tegen wat geacht wordt de oorzaak te zijn, in plaats van een maatregel tegen het werkelijke probleem.

Er zijn vier manieren om te controleren of de meest waarschijnlijke oorzaak ook de werkelijke oorzaak is: feitenverificatie, observatie, onderzoeksverificatie en resultatenverificatie. Elk van deze onderdelen vraagt om een verschillende aanpak:

Feitenverificatie: Onderzoek alle veronderstellingen en aannames. Als blijkt dat de veronderstellingen of aannames op feitelijkheid berusten, is het goed mogelijk dat de werkelijke oorzaak is gevonden.

Observatie: Geef uw ogen goed de kost. Soms kan de werkelijke oorzaak worden aangetoond door alleen maar te kijken hoe de oorzaak het probleem heeft veroorzaakt.

Onderzoeksverificatie: Experimenteer om te kijken of het probleem zich herhaalt wanneer de meest waarschijnlijke oorzaak wordt toegepast. Wanneer het probleem te allen tijde kan worden opgeheven of herhaald, is de werkelijke oorzaak aangetoond.

Resultatenverificatie: Probeer de storing op te heffen maar houd de situatie goed in het oog en kijk of het probleem zich niet langer meer voordoet. Houd de situatie na het oplossen van het probleem altijd nauwlettend in de gaten want anders is het nog niet zeker of het allemaal goed blijft werken.

Welk verificatiesysteem moet worden toegepast? Bekijk wat het goedkoopst, het makkelijkst, het snelst en het veiligst is, en begin daar dan mee.

Als de meest waarschijnlijke oorzaak niet geverifieerd kan worden, ga dan terug naar de lijst met mogelijke oorzaken en kies de volgende meest waarschijnlijke oorzaak, die dan vervolgens op waarheid onderzocht moet worden.

IMPLEMENTATIE ATS IN ORGANISATIES

Om ATS succesvol te implementeren en verankeren in organisaties zijn een drietal 'bouwstenen' nodig:

1. Vaardigheidstraining van medewerkers in de methode.
2. Het aanpassen van incident- en problemmanagementprocessen.
3. Het creëren van een effectieve werkomgeving.

Met name over de laatste twee bouwstenen willen wij in deze paragraaf graag enige toelichting geven.

Aanpassen processen voor incident- en problemmanagement

In vrijwel iedere organisatie is er sprake van een al dan niet formele wijze van afhandeling en oplossing van klantproblemen, al dan niet 'Incident-' en 'Problemmanagement'-processen genoemd. Indien men ATS succesvol wil implementeren is het wenselijk om deze processen zodanig aan te passen dat ProbleemAnalyse uiteindelijke ingebed is in deze processen.

Om dit te realiseren kunnen doorgaans de volgende stappen worden doorlopen:

1. Beschrijf het incident- of problemmanagementproces (als dat al niet conform de actuele situatie op papier stond).
2. Zoek naar kritische punten in deze processen waarbij het gebruik van ProbleemAnalyse tot de nodige verbetering kan leiden. Typische punten waar veelal veel winst valt te behalen zijn de 'handovers' of momenten van 'dispatchen' tussen de ene en de volgende lijn van troubleshooters. De informatieoverdracht kan immers sterk worden verbeterd als deze, conform de IS en IS-NIET specificatie compleet, accuraat en feitelijk geschiedt.
4. Identificeer heldere 'triggers' voor het gebruik van (eventueel delen van) ProbleemAnalyse. Het is uitermate belangrijk dat troubleshooters een helder signaal

hebben wanneer men delen van deze methode moet gebruiken en wanneer dat (nog) niet hoeft. Zeker bij aanvang van implementatie zal anders blijken dat het expliciet gebruiken van de methode te weinig of te laat gebeurd.

5. Pas de incident- en problemmanagement-processen zodanig aan dat de triggers en het gebruik van de methoden in de processen zijn opgenomen.

In figuur 5 is een voorbeeld gegeven van een proces met meerdere niveaus van troubleshooting waarin het gebruik van Probleem-Analyse is ingebed. Deze manier maakt het tevens mogelijk om op bepaalde momenten effectief te communiceren naar de klant.

Creëren van effectieve werkomgeving

Het aanpassen van werkprocessen is reeds een belangrijke voorwaarde om een succesvolle implementatie van ATS te realiseren. Er zijn echter meer mogelijkheden om een effectieve werkomgeving te creëren, een omgeving waarin troubleshooters in alle opzichten gestimuleerd worden tot het gebruik van de methoden bij het oplossen van problemen (zie ook Vonk, 2003). We noemen er een aantal:

- informatiesystemen die op het gebruik van ProbleemAnalyse zijn aangepast;
- heldere geformuleerde doelstellingen voor het gebruik van ATS;
- het opnemen van de gewenste vaardigheid in ATS in functiebeschrijvingen alsmede het meenemen van de werkelijke aangetoonde vaardigheid bij evaluatiegesprekken;
- managers die het gebruik stimuleren en zo mogelijk een voorbeeldfunctie vervullen

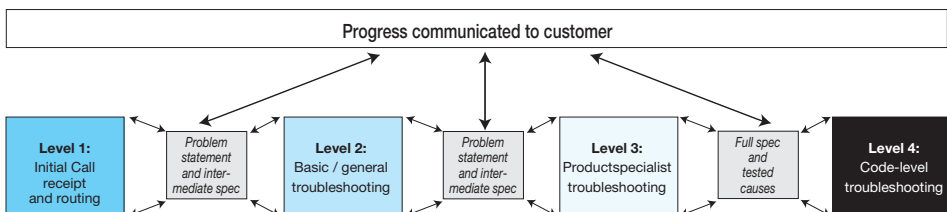
door zelf continu 'procesvragen' in plaats van 'inhoudelijke' vragen te stellen;

- publicatie van 'war-stories';
- beschikbaarstelling van oefencases op het intranet.

RESULTATEN IN DE PRAKTIJK

In de inleiding van dit artikel werd reeds opgemerkt dat de methode van Probleem-Analyse zich aan het ontwikkelen is als een de facto standaard. De tot nu toe behaalde resultaten bij een groot aantal global providers zijn hier mede debet aan. Een aantal voorbeelden:

- Een wereldwijde customer support organisatie gebruikte ATS als de standaard methode voor het afhandelen van geëscalereerde klantproblemen. Resultaat: de totaal benodigde tijd voor het oplossen van alle onderhanden problemen werd gereduceerd met 16000 dagen.
- Een fabrikant van telecom producten adopteerde ATS om klantproblemen af te handelen. Resultaat: de gemiddelde tijd om een klantklacht af te handelen reduceerde met 58%.
- Een producent van elektronische apparatuur gebruikte ATS in haar call center. Resultaat: De verbeterde diagnoses verminderde het aantal 'on-site' bezoeken door engineers met 41%, met honderden duizenden dollars kostenbesparing als gevolg.



Figuur 5 Voorbeeld van een Probleemgestuurd support model

VOORBEELD UITGEWERKTE PROBLEEMANALYSE

Inleidende tekst

Vevo B.V.

Vevo B.V. is producent van complete computersystemen voor het monitoren en beveiligen van gebouwen, terreinen of productieprocessen. Centraal onderdeel vormen de DKK-40 units, welke inmiddels van een tweede, verbeterde, generatie zijn. Er zijn verschillende modellen van deze unit beschikbaar die gemakkelijk kunnen worden aangepast aan de wensen van de klant (DKK-41 voor wooncomplexen, DKK-42 voor petrochemische industrie, etc.).

Er is zojuist bij Vevo weer een foutmelding 'memory protection error' binnengekomen van Klant A, een grote klant welke diverse van onze systemen in gebruik heeft. De error message gaat gepaard met het volledig plat leggen van het systeem. Het probleem is inmiddels al meerdere malen opgetreden, niet alleen bij Klant A, maar ook bij andere klanten. Meestal werd het probleem snel door Vevo afgehandeld door het vervangen van een printplaat, maar op sommige units bleven de fouten terugkomen.

In de matrix op de volgende pagina's vindt u een volledig uitgewerkte ProbleemAnalyse van deze case.

LITERATUUR

- **Kepner, C.H. and B.B. Tregoe.** *The new rational manager*, Princeton, 1997.
- **Visser, B. en R. Vonk.** 'Analytic Trouble Shooting'. In: *Maintenance Management*, nummer 4, 2001.
- **Vonk, R.** 'Succesvol trainen van medewerkers (deel 1 en deel 2)'. In: *Maintenance Management*, nummer 6 en 7, 2003.

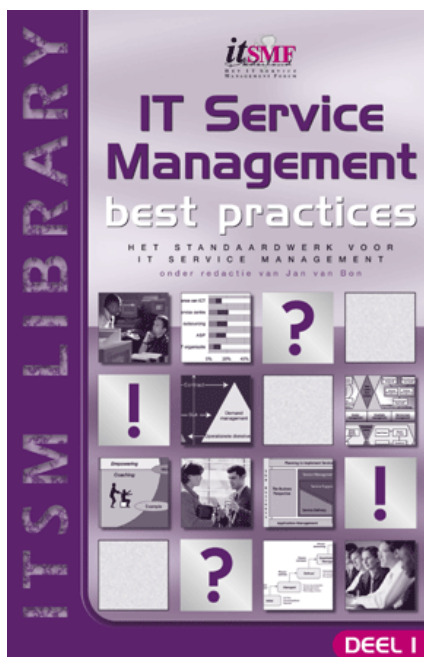
PROBLEEMANALYSE MATRIX

408

Beschrijf het probleem			Formuleer mogelijke oorzaken
Geef probleemomschrijving (geef een object en een afwijking) DKK's geven foutmelding "memory protection error"			... via kenmerken en veranderingen, of...
Specificeer het probleem			Wat is anders, uniek, apart, kenmerkend aan elke IS in vergelijking tot de bijbehorende IS NIET?
WAT	IS	IS NIET	Kenmerken
Welk ding/object?	DKK-42 en DKK-44	DKK-41 en DKK-43	Andere bevestigingsmethode CPU
	Huidige (tweede) generatie	Vorige generatie	Andere verbinding voeding
	Recente DKK-42's : serie 10823 en hoger	DKK-42 serie < 10823	Tweede generatie is geavanceerder
	Recente DKK-44's : serie 5589 en hoger	DKK-44 < 5589	
Welke afwijking?	Foutmelding "Memory protection error"	Foutmelding "I/O violation"	
WAAR			
Waar geografisch?	Door het hele land	Lokaal, enkele regio's	
	Klant A	Klant C	
	Klant B	Klant D	
	Andere klanten in de Randstad	Klant E	
Waar op het object?	?? (meer gegevens nodig)	?? (meer gegevens nodig)	
WANNEER			
Wanneer voor het eerst?	20 januari 2004	Daarvoor	
Wanneer sindsdien?	Op willekeurige data, met tussenpozen	Periodiek of continu	
Wanneer in de levenscyclus?	Tijdens gebruik bij de klant	Tijdens productie, bij eindcontrole en testfase bij Vevo, Bij levering en installatie bij klant	
OMVANG			
Hoeveel objecten?	34 DKK-42's en 17 DKK-44's	meer/minder	
Hoe groot is de afwijking?	-- (niet van toepassing)	-- (niet van toepassing)	
Hoeveel afwijkingen per object?	72 foutmeldingen bij 51 DKK's	Meer/minder	
Wat is de trend?	Toenemend	Stabiel of afnemend	



"*Analytic Trouble Shooting®*, *The best practice for problem solving*" is gepubliceerd in deel 1 van **IT Service Management, best practices**, in de ITSM Library van ITSMF-Nederland.



Ieder nieuw deel in deze reeks wordt jaarlijks samengesteld onder auspiciën van de ITSM Redactieraad, en gepubliceerd tijdens het ITSMF voorjaarscongres "Best Practices in IT Service Management".

De uitgave is verkrijgbaar via ITSMF-Nederland.

Meer informatie:

ITSMF voorjaarscongres "Best Practices in IT Service Management":

<http://www.best-practices-in-ITSM.nl>

Boekinformatie: IT Service Management, best practices, deel 1

<http://nl.itsmportal.net/goto/literatuur/boek/136.xml>

Boekenreeks ITSM Library:

<http://nl.itsmportal.net/goto/literatuur/boek/204.xml>