




Redactioneel

Het is alweer de zesde jaargang van TNN. De redactie hoopt dat er dit jaar nog meer bijdragen van onze leden komen voor deze nieuwsbrief en onze site. Er zijn plannen genoeg voor het komende jaar. In ieder geval de redactie uitbreiden met nog wat mensen voor het bijhouden en verwerken van congressen en cursussen. Verder willen wij het aantal columnisten, nu alleen nog Erik van Veenendaal, uitbreiden. U ziet, de redactie probeert constant om TNN op een hoger kwalitatief niveau te brengen. Wij hopen dan ook dat u weer veel leesplezier aan deze nieuwsbrief zult beleven. 

In dit nummer

Redactioneel	1
Van de Voorzitters	1
Van de Secretaris	1
Kwaliteitszorg in ICT-projecten	2
Verslag Thema-avond 13 december 2001	3
TA - Test data management	4
TA - Automatisering van testmanagement	5
TestNet Column	
Hulpmiddelen voor non-functional testen	5
EuroSTAR 2001 Verslag	6
EuroSTAR 2001 Paper	7
Verschijningsdata	13
Kortingsacties	13
Ingezonden stuk	13
Evenementen	14
Colofon	14
Kortingsformulier	15

Van de voorzitters


Door Martin Pol en
Hans van Loenhoud

Een (be)vingrijjk 2002!

Namens het bestuur van TestNet wensen wij u allen een gezond en succesvol 2002 toe. Moge u een goede balans vinden tussen uw leven en uw werk in het uitoefenen van ons mooie testmetier.

2002 wordt voor TestNet een bijzonder jaar, waarin we ons opmaken voor het eerste lustrum. Vijf jaar een stevig platform voor testend Nederland, tenminste voor u als leden en voor hen die TestNet anderszins kennen. Ondanks de vele aankondigingen in presentaties, brochures en bij gelegenheid ook perspublicaties en ondanks TNN en onze website hebben wij het idee dat we een aantal testcollega's nog niet bereiken. Dat is niet goed! Daarom, zegt het voort, zegt het voort: Er bestaat een vereniging voor testers, een club die zich met hart en ziel inzet om de Nederlandse testwereld te verenigen, internationaal bruggen te slaan met zusterorganisaties, kennis over het mooie testvak met elkaar te delen en zelf nieuwe concepten te ontwikkelen en creatieve oplossingen te bedenken: ons aller onvolprezen TestNet!


TestNet heeft bewezen, in deze bij haar oprichting gedefinieerde doelstellingen te kunnen voorzien. Hoe meer we erin slagen alle 'brothers in arms' te rekruteren, des te beter

het lukt om onze doelstellingen te bereiken en des te meer voordeel en plezier we allen aan ons lidmaatschap beleven. Laten we elkaar helpen om TestNet te doen groeien en bloeien door nieuwe leden de weg naar onze vereniging te wijzen. Veel succes daarbij toegewenst en dank voor uw hulp. 

Van de secretaris

Door Marco Jansen van Doorn

Terugblikkend op het afgelopen jaar is ook 2001 weer een succesvol jaar geweest wat betreft het ledenaantal van onze vereniging. Startten we het jaar met 530 leden, de eindstand was 620 leden. Dit is toch weer een toename van 17%. Opvallend is dat dit percentage identiek is aan de toename in het jaar 2000. Ook al is het gevaarlijk op basis van de relatief weinig gegevens die er over het ledenaantal van onze vereniging zijn (we bestaan dit jaar 5 jaar) trends te willen signaleren, lijkt de aanwas zich te stabiliseren. Interessant (voor mij althans) te volgen of dezelfde groei ook dit jaar weer zal optreden.

Groei heeft op zich een positieve uitstraling. Wel zullen we ons moeten beseffen dat een groeiend ledenaantal betekent dat de inspanning om die leden te bedienen ook zal groeien. Heeft u interesse om hier aan mee te werken, op wat voor gebied dan ook, schroom niet om dit aan te geven door een e-mail te sturen: secretaris@testnet.org. 

Kwaliteitszorg in ICT-projecten

Door Drs. H.J.J. Cannegieter
SYSQA

De laatste jaren neemt de interesse in kwaliteitszorg in ICT-projecten toe. Veel organisaties zien in dat het bewaken van de kwaliteit door middel van testen duur is; kwaliteitsmaatregelen eerder in het systeemontwikkeltraject lonen! Er bestaan ook diverse methodes voor het definiëren, borgen en verbeteren van de kwaliteit in ICT-projecten. Voorbeelden zijn ISO 9126, Inspecties en verbetercycli. Tot voor kort bestond nog geen integrale methode voor kwaliteitszorg in ICT-projecten. Die bestaat sinds kort wel: PROQA. PROQA staat voor PROject Quality Assurance en is ontwikkeld door een lid van TestNet. In dit stuk wordt de methode verder toegelicht aan de hand van de fases.

samenhangende set maatregelen gemaakt die elkaar versterken.

PROQA bestaat uit vijf fases:

- Kwaliteitsdefinitie;
- Kwaliteitsborging;
- Kwaliteitsverbetering;
- Testen;
- Afronding.

De fase *kwaliteitsdefinitie* omvat onder andere het inventariseren van de stakeholders en kritische succesfactoren. Hier worden de eisen aan eind- en tussenproducten uit afgeleid. De eisen aan het eindproduct vormen direct de acceptatiecriteria die weer input vormen voor de teststrategie.

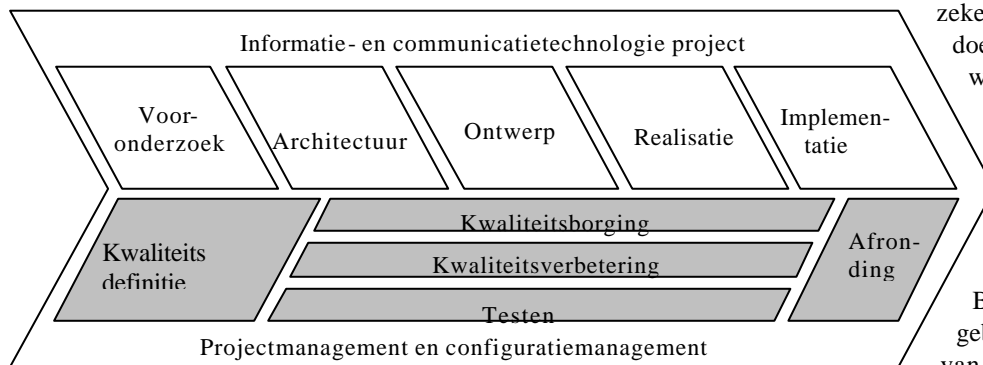
Een andere belangrijke set activiteiten, die wordt uitgevoerd tijdens de kwaliteitsdefinitie, heeft betrekking op risicomangement. Om het project succesvol tot een einde te brengen is het belangrijk dat helder is welke factoren een negatieve invloed kunnen hebben op het eindresultaat.

wat de risico's zijn, kan vastgesteld worden op welke wijze het project wordt aangepakt. Anders gezegd; de projectprocessen kunnen bepaald worden. Diverse organisaties kennen een standaard aanpak van projecten, dikwijls vastgelegd in het kwaliteitssysteem. Dit is het moment om te beoordelen of deze aanpak ook voor dit project de beste aanpak is. Afwijken mag, als je projectdoelstellingen het maar legitimeert en de afwijking beheerst gebeurt. Met dit laatste wordt bedoeld dat er de alternatieve werkwijze bij afwijking ook beschreven en controleerbaar dient te zijn. De kwaliteitsdefinitie wordt afgesloten met het inrichten van de rest van het kwaliteitstraject en het opstellen van het kwaliteitsplan.

Tijdens de uitvoering van het project vindt de kwaliteitsborging plaats. Doel van kwaliteitsborging is zekerstellen dat de beoogde doelstellingen gerealiseerd worden. Dit gebeurt door de kwaliteit van tussenproducten te beoordelen en eventuele fouten zo vroeg mogelijk te herstellen.

Bij kwaliteitsborging kan gebruik worden gemaakt van bovenstaande technieken, die in het boek nader worden behandeld. De derde fase van PROQA is *kwaliteitsverbetering*. Kwaliteitsverbetering begint met kwaliteitsadvisering. Kwaliteitsadvisering heeft betrekking op het ter beschikking stellen en optimaliseren van technieken, hulpmiddelen, normen, standaards en dergelijke. De

TestNet Nieuws



Afbeelding 1: PROQA model

PROQA is een methode voor het definiëren, borgen en verbeteren van kwaliteit in ICT-projecten. PROQA maakt gebruik van methoden en technieken die hun sporen hebben bewezen en voegt hier het één en ander aan toe. Hiervan wordt een

Hiervoor worden de risico's geïnventariseerd, bepaald of er maatregelen worden genomen naar aanleiding hiervan en wordt de ontwikkeling van deze risico's gevolgd tijdens het project.

Nu bepaald is wat het einddoel van het project is in termen van kritische succesfactoren en acceptatiecriteria en bepaald is



teamleden hebben over het algemeen diverse zaken ter beschikking die hun werk ondersteunen. Met het ter beschikking stellen en optimaliseren van het gebruik ervan kan reeds veel winst worden geboekt. Gebruik eerst wat al bedacht en beschikbaar is!
De tweede hoofdactiviteit van kwaliteitsverbetering heeft betrekking op het verbeteren

sjabloon voor een kwaliteitsplan opgenomen. Aan de hand van PROQA kan kwaliteitszorg in een ICT-project dus worden ingericht en uitgevoerd.

Enige tijd geleden zei een opdrachtgever eens tegen mij dat testen zo duur is en er altijd ongewenste verrassingen uit het testtraject kwamen. Als repliek gaf ik dat

bedrijf SQS/Aqsoft. SQS is een van oorsprong Duitse firma waarvan weleens wordt gezegd dat wat Iquip en CMG voor Nederland is, SQS is voor Duitsland. Aqsoft is daarentegen is een Nederlands bedrijf maar dat recentelijk door SQS is overgenomen.

De avond begon met een presentatie van het testprocesmodel zoals SQS dat hanteert. Zie afbeelding 2

Het grootste deel van de avond werd ingeruimd voor een brainstormsessie. In verschillende

groepen gingen we uiteen met de opdracht ééns na te denken over wat wij als testers graag van het totale testproces geautomatiseerd zouden willen zien. Doel hiervan is testtoolleveranciers inzicht te geven in de wensen uit het veld.

Iedere discussiegroep kreeg vervolgens één onderwerp mee afkomstig uit het genoemde procesmodel. De groep waar ik deel van uitmaakte, mocht "dromen" over het aandachtsgebied "Testcase Specification". Dat het dromen voor ons testers best moeilijk is, bleek al snel. Ideeën werden direct getoetst op haalbaarheid om snel in de prullenmand te verdwijnen in plaats van op de flip-over. In mijn rol van moderator heb ik gelukkig nog enkele van de ideeën van de ondergang kunnen redden.

Het resultaat van onze sessie was:

1. Het beschikbaarstellen van een template/wizard per testspecificatietechniek.

Borgingstechnieken	Formeelheid	Intern / extern	Uitgevoerd door
Collegiale review	Zeer informeel	Intern	Collega
Expert review	Informeel	Intern	Expert, intern of extern
Management review	Redelijk formeel	Intern	Expert, namens het management
Structured walkthrough	Redelijk formeel	Intern	Groep betrokkenen
Inspectie	Formeel	Intern	Groep betrokkenen
Audit	Zeer formeel	Extern	Onafhankelijke functionaris van buiten het project

Tabel 1: Borgingstechnieken

van processen en hulpmiddelen. Gedurende het project blijkt soms dat processen niet optimaal lopen en kunnen andere verbeteringen worden doorgevoerd. Hiervoor wordt gebruik gemaakt van een verbetercyclus.

In ieder kwaliteitstraject vindt aan het einde van de productie een kwaliteitscontrole plaats. Vandaar dat *testen* de vierde PROQA-fase is. Tijdens het testen wordt vastgesteld of het eindproduct aan de in de, kwaliteitsdefinitie vastgestelde, eisen voldoet.

Het kwaliteitstraject wordt afgesloten met de fase afronding. De afronding bestaat uit conservering van projectdocumentatie en het uitvoeren van een projectbrede evaluatie.

De verschillende fases en activiteiten, alsmede de gebruikte technieken worden in het boek behandeld. Ook zijn een aantal checklists en

kwaliteitsmaatregelen ook eerder in het project genomen dienen te worden. Hij was het er volledig mee eens, dat verhaal was bekend. Hij vroeg zich alleen af hoe dat moest. PROQA geeft hier invulling aan.

Kwaliteitszorg in ICT-projecten. De PROQA-methode.
ISBN 90-440-0369-0



Verslag TestNet themavond van 13 december 2001

Door: Egbert Egberts

Dit was alweer de laatste TestNet themavond van dit jaar en stond deze keer in het thema van Testautomatisering. De belangstelling was groot; zo'n 60 leden gaven acte de presence!

De avond werd deze keer volledig verzorgd door het



- Hierdoor ontstaat een uniforme manier van vastleggen en verkleint de kans op fouten;
2. Automatische generatie van testgevallen vanuit (wel formeel beschreven)

- een inventarisatie van de belangrijkste wensen:
7. Ondersteuning bij de overgangen requirements -> testgevallen -> testdata
 8. Ondersteuning bij de inrichting van een

TA - Test data management

Door Wim van Leunen

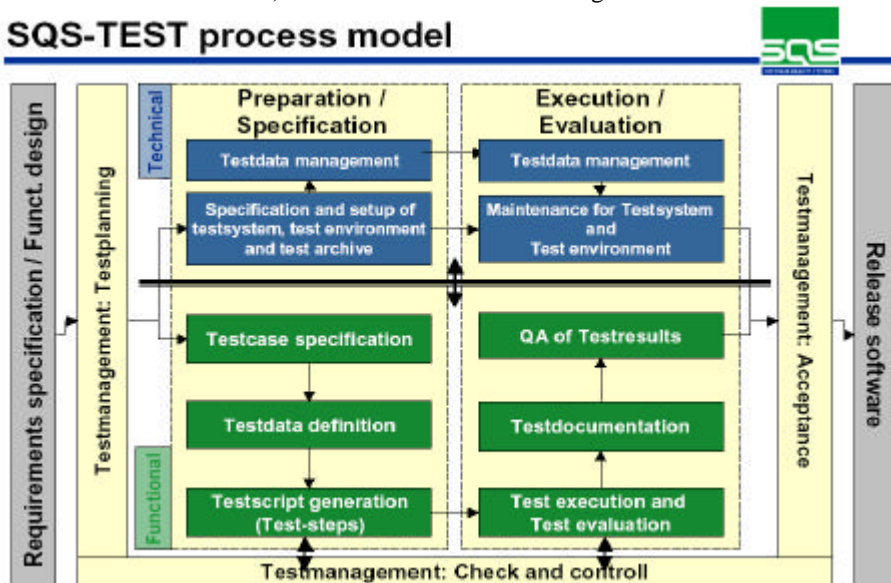
Er worden 3 soorten testdata onderscheiden:

- Initiële data (B.v. in een database of spreadsheet);
- Invoerdata
- Uitvoerdata (inhoud database na de test)

Per test datatype zijn problemen mbt. data management opgesomd, wensen geuit en soms zelfs oplossingen aangedragen:

- Initiële data
 - a) Opbouw :
 - d.m.v. testtools kan het automatisch vullen van een database plaatsvinden
- DB manipulatiertools zorgen ervoor dat de data geselecteerd, gewijzigd en gecontroleerd kan worden. De meeste testtoolleveranciers leveren hier hulpmiddelen voor.
- Onderhoud: Een wenst is om een tool te gebruiken die op basis van parameters een database kan aanmaken en dit voor verschillende platformen.
- Houdbaarheid: Afhankelijk van veranderingen in de applicatie of testware zal de test data moeten worden aangepast. Mogelijke oplossingen zijn:
 - a) Eerst een regressietest doen en dan de resultaten bekijken;
 - b) Eerst de impact bekijken op de test data

SQS-TEST process model



Afbeelding 2: Test process model

3. Rapportages vanuit de testspecificaties zoals inzicht in dekking op logisch niveau;
4. Automatische testdatagenerator vanuit o.a. de logische testspecificaties;
5. Geautomatiseerde ondersteuning bij de transformatie van logische naar fysieke testgevallen;
6. Het automatisch bijhouden van de tijd die nodig is om testspecificaties op te stellen (planningsinstrument).

9. Ondersteuning bij de beantwoording van de vraag: wanneer zijn we klaar met specificeren en wanneer zijn we klaar met testen.

Tijdens de avond hebben we ook twee keer een demonstratie gegeven van SQS-Test, dit is de oplossing voor testautomatisering van SQS.

Al met al een interessante avond vooral ook om op deze wijze kennis te maken met een wat minder bekende leverancier van testdiensten.



In de aansluitende plenaire zitting werd per groep de voorstellen toegelicht en was er gelegenheid tot het stellen van vragen.

De avond werd afgesloten met

en dan de testware aanpassen, vervolgens een regressietest uitvoeren.

- **Invoerdata**
Een probleem met de invoerdata is dat deze op gespannen voet kan staan met de initiële data. Bij het invoeren van fysieke waarden, b.v. rekeningnummers, kan het voorkomen dat deze al in de database aanwezig zijn. Een oplossing hiervoor is om de invoerdata in een spreadsheet vast te leggen.
- **Uitvoerdata:**
De uitvoerdata kan gecontroleerd worden door de recorded data en de expected data met elkaar te vergelijken. TestFrame biedt hier mogelijkheden voor, maar ook de meeste testtools hebben deze functionaliteit. 

TA – Automatisering van testmanagement

Door Marco Jansen van Doorn

De groep die zich op de thema-avond ‘automatisering van het testproces’ boog over testmanagement, heeft zich de vraag gesteld wat we van een applicatie zouden verlangen die het managen van een testtraject zou ondersteunen. Deze hypothetische applicatie zou bepaalde informatie met één druk op de knop moeten kunnen produceren. We kwamen daarbij uit op de volgende vijf wensen:

- **Resource Planning & Tracking**
Snel inzicht in alle (nog) uit te voeren taken, de menskracht die daarvoor nodig zou zijn en de expertises die deze medewerkers zouden moeten hebben is op elk

moment tijdens een testtraject bijzonder waardevol. Dit helpt bij het plannen, en achteraf zou het inzicht geven in die processen die efficiënter kunnen.

- **Risicoanalyse**
Als we naar aanleiding van het ontwerp van het systeem (in welke vorm dan ook) direct kunnen zien welk deel van het systeem een hoge faalkans heeft, wordt het bepalen van de risico’s (= faalkans * schade) vergemakkelijkt.
- **Impactanalyse**
Inzicht in de gevolgen van wijzigingen in een applicatie, zowel voor de uit te voeren testgevallen (welke moeten we minimaal nog een keer uitvoeren) als het onderhoud van testware (welke testgevallen moeten aangepast worden), helpt ons bij de testplanning. Dit geldt niet alleen voor het plannen van een testtraject in een onderhoudssituatie, maar voor elk testtraject, want gedurende het testproces zullen naar aanleiding van reeds uitgevoerde tests nieuwe releases worden opgeleverd.
- **Testdekking**
Op elk moment in het testtraject willen we inzicht in de actuele testdekking, zowel voor de specificatie (hoeveel testgevallen zijn er gespecificeerd ten opzichte van het gewenste aantal testgevallen) als voor de uitvoering (welk deel van de gespecificeerde testgevallen zijn reeds uitgevoerd). Deze testdekking moet gekoppeld worden aan de risicoanalyse om te kunnen

bepalen welk risico we lopen als we de huidige testfase af zouden sluiten.

- **Metrics**
Het ideale testmanagementtool bouwt niet alleen metrics op, maar gebruikt deze ook. Bij nieuwe trajecten willen we tenslotte kunnen putten uit de reeds opgedane ervaringen.

Als dit allemaal gerealiseerd zou zijn, kan al deze informatie gebruikt worden om ons twee getallen te geven: de kosten die verwacht worden als we doorgaan met testen en de kosten die verwacht worden als we zouden stoppen met testen (door gebrek aan kwaliteit in productie). Uiteindelijk wil een manager die optie kiezen die de laagste kosten geeft. 

TestNet Column

Alweer de tweede column. Deze keer een stuk over de ISO 9126 standaard, die voor ons als tester van groot belang is. Voor vragen of een reactie kunt u mailen met Erik van Veenendaal

Hulpmiddelen voor “Non-functional” testen

Erik van Veenendaal
eve@improveqs.nl

Bij het testen van software producten wordt veelal verwacht dat niet slechts de functionaliteit maar ook allerlei zogeheten niet-functionele kwaliteitskarakteristieken worden getest. Alhoewel in een testplan veelal zaken zoals onderhoudbaarheid of bruikbaarheid worden aangeduid als zijnde belangrijk, is het voor testers vaak moeilijk om hieraan concreet inhoud te geven. Testaanpakken zoals TMap en TestFrame leveren op dit gebied slechts een beperkte


bijdrage. In TMap zijn wel een groot aantal checklists te vinden die ondersteuning bieden bij het beoordelen of de noodzakelijke maatregelen in het kader van een bepaalde kwaliteitskarakteristiek zijn genomen; het zogenaamde indirect testen. Er zijn natuurlijk bronnen die specifiek aandacht geven aan één aspect; bijv. performance testen. Er is echter behoefte aan een samenhangend geheel waarin aan de verschillende aspecten van kwaliteit aandacht wordt geschonken. In dit kader dienen twee belangrijke ontwikkelingen te worden besproken die testers kunnen helpen bij non-functional testing.

De ISO-commissie heeft een internationale standaard gedefinieerd (ISO9126) die een eenduidige definitie biedt van de beschikbare kwaliteitskarakteristieken. ISO onderkend 6 hoofdkarakteristieken (functionaliteit, efficiency, bruikbaarheid, onderhoudbaarheid, portabiliteit en betrouwbaarheid) en 21 subkarakteristieken. (In bijlage B van de 2^e druk van TMap is hierover meer informatie te vinden.) Recentelijk heeft ISO de 9126-standaard uitgebreid met een tweetal delen die nu in concept beschikbaar zijn, nl. ISO 9126-2 en 9126-3. Deze delen zijn voor testers zeer interessant en bieden een grote set metrieken voor elke kwaliteitskarakteristiek. Oftewel hoe kunnen de aspecten zoals onderhoudbaarheid en betrouwbaarheid concreet worden gemaakt. Deze metrieken kunnen door de tester bijvoorbeeld worden gebruikt als acceptatiecriteria en onderdeel uit gaan maken

van de periodieke risicorapportage. Deel 2 is gericht op de ontwikkelaar en deel 3 op de "black-box" tester. Een voorbeeld van een toepassing van deze metrieken in de praktijk is beschreven in het artikel "Measuring Software Product Quality during Testing" te vinden op www.improveqs.nl

Een tweede belangrijke ontwikkeling is de totstandkoming van een aantal standaards voor non-functional testen. Diverse werkgroepen binnen de British Computer Society (BCS) zijn hier reeds geruime tijd mee bezig en definiëren documenten waarin wordt aangegeven hoe een bepaalde kwaliteitskarakteristiek kan worden getest. Inmiddels zijn er al conceptstandaards van reliability, usability en maintainability beschikbaar (www.testingstandards.co.uk). In de usability test standaard wordt onder andere ingegaan op technieken zoals heuristic evaluation, SUMI en cognitieve walkthrough. In de nabije toekomst zullen ook documenten voor onder andere security, recovery, performance en interoperability testen worden opgeleverd. Bruikbare en praktische documenten, welke zijn opgesteld door praktijkmensen. In het kader van het ISEB testcertificatie programma (zie de vorige TestNet Nieuws) zullen deze standaards uitgebreid aan de orde komen op het Practitioner niveau.

Al met al ontwikkelingen die ons helpen om onze klanten beter te kunnen bedienen. We kunnen met behulp van ISO 9126 het begrip kwaliteit beter concretiseren en meetbaar maken. Met de BCS non-

functional test standards kunnen we daadwerkelijk aan de slag met testtechnieken. Een nieuwe uitdaging voor de test professional !! 

EuroSTAR 2001

Door Annemarie van Gelder
Kadaster

Inleiding

Dankzij mijn collega's (Jaap en Jan), een prijs van Testnet en een gesponsorde vlucht inclusief overnachting, was het voor mij mogelijk een bezoek te brengen aan EuroSTAR 2001, the 9th European International Testing Conference, in Stockholm.

De trip begon op zondag 18 november, veel te vroeg. Aangezien zelfs de NS op dit tijdstip nog geen diensten verleent, werd de hele Kadaster delegatie, bestaande uit Jan Marlies en mijzelf, per auto naar Schiphol gebracht.

Enmaal op Schiphol aangekomen werden we snel geconfronteerd met een andere tester, namelijk Meile (waarvan ik later begreep dat dit zijn voornaam was). Net als wij was hij voornemens tijd te besteden aan het bezichtigen van Stockholm. Nadat we geland waren in een zonnig Zweden bleek mijn bagage niet meegekomen te zijn. Een toilettaas met een tandenborstel, kam en maar liefst 2 busjes scheerschuim werd als bij wijze van excuus door de KLM aangeboden. Jan stond direct toe een nieuwe garderobe aan te schaffen, dat natuurlijk geheel uit de oranje testkleur moest bestaan.

Stockholm

Stockholm was zeer de moeite waard voor een wandeling. De hoofdstad bestaat uit 14 eilanden, 40 bruggen en heeft ongeveer 650.000 inwoners. Het oudste deel van de stad is gevestigd op het eiland Gambla Stan. Dit deel bestaat uit kleine gezellige straatjes, waar je makkelijk kan verdwalen, maar telkens weer op dezelfde straat uit komt. Stockholm heeft vele mooie oude gebouwen zoals de opera, het paleis en het stadhuis. Kortom een aanrader!

Congres

Maandagochtend begon het serieuze deel van de week. Een week vol met tutorials, exhibitions en tracks stond ons te wachten. De inhoud van het congres is heel interessant en divers en hiervoor verwijs ik jullie naar de site:

www.testingconference.com

In dit artikel beperk ik me tot de meest opvallende zaken met betrekking tot de conferentie.

Ten eerste een opmerkelijke titel waarmee dit congres wordt aangeduid, "Let's all work together". Wie zijn "all" en wanneer moeten we "work together"?

Al redelijk vroeg in de week merkte ik dat alleen testers op het congres aanwezig waren en dan met name testmanagers en testconsultants en een enkele testengineer. Waar bleven alle managers? Enige support van hun kant zou het testen op een hoger niveau kunnen brengen. Kan hun afwezigheid een indicatie geven van de positie van testen op hun prioriteitenlijstje?

En dan de ontwikkelaars, bij welke track waren zij aanwezig? In ieder geval niet bij diegene die ik gevolgd heb. Tijdens het congres werd er (bijna) tot vervelendst toe gewezen op de belangrijkheid

van de samenwerking tussen testers en ontwikkelaars. Alle testers knikten bevestigend, natuurlijk moeten we samenwerken. Helaas is het nog steeds zo dat een samenwerking pas tot stand kan komen als beide partijen bevestigend kunnen knikken. Dus volgend jaar toch maar samen met collega's ontwikkelaars het congres bezoeken?


Verder viel me op dat het noorden van Europa volop vertegenwoordigd was. Het zuiden, met landen zoals Spanje, Frankrijk, Griekenland liet het behoorlijk afweten. Is het mogelijk dat de geografische situatie van het congres een reden hiervoor is of heeft het testen in het zuiden van Europa nog geen plaats op de kaart van de testwereld verworven.

Slot

Tot slot viel me uiteraard op dat er een hoop te leren viel op het EuroSTAR. Onderwerpen als education, metrics, testtools, test process, internet, test techniques, risks, etc. zijn volop aan bod gekomen. Elk uur was er de mogelijkheid om uit minimaal vier tracks te kiezen en de meeste interessante te bezoeken. Het toepassen van testen in de verschillende organisaties heeft bij mij tot nieuwe ideeën over testen geleid, die zeker tot verbetering in het testproces kunnen leiden.

Vrijdagavond gingen we weer naar huis toe, het was tijd geworden om flink bij te slapen. In het vliegtuig dacht ik nog eens na over het congres en voor mij kon ik al snel concluderen dat het een boeiend, interessant congres was waarbij we all have

worked together.

Mijn collega's wil ik tot slot nog adviseren volgend jaar vooral naar de TestNetavond te gaan, zodat ook zij een kaartje voor het volgende congres in Engeland kunnen winnen. 

EuroSTAR paper

Wanneer er meer mensen zijn die een artikel voor een conferentie hebben ingediend, dan kan dit aan TNN worden aangeboden. Hierdoor kunnen de leden van TestNet kennis nemen van de expertise die binnen onze vereniging aanwezig is. Tijdens de laatste EuroSTAR in Stockholm heeft Gerard Numan een presentatie gegeven over chain testing.

Testing a chain of applications

Door Gerard Numan

Chain Testing

DEFINITION OF A CHAIN

A chain is a system of several applications communicating with one another. A chain in this sense is not just a collection of applications but it acts as a system in itself, a body. Like a body, each part only makes sense as a part of the whole. And, as in a body, each part will be highly dependent on the interaction with other parts. Not only do the veins of each part have to fit with the veins in other parts, the blood that is streaming through the veins should be in the right amount, constitution, temperature and quality as expected by the other body part.

Like a body there should be a central intelligence, a brain, coordinating the overall processes from the overall purpose of the whole system. Depending on 'intelligence'

shared by applications, a chain becomes more complicated, becomes more 'chain-like' so to speak.

A process that is cut up into different sub-processes being executed in different applications will have a chain-like character and requires chain testing. Depending on the interdependence, changes in one process might even imply chain testing (regression) of other processes as well.

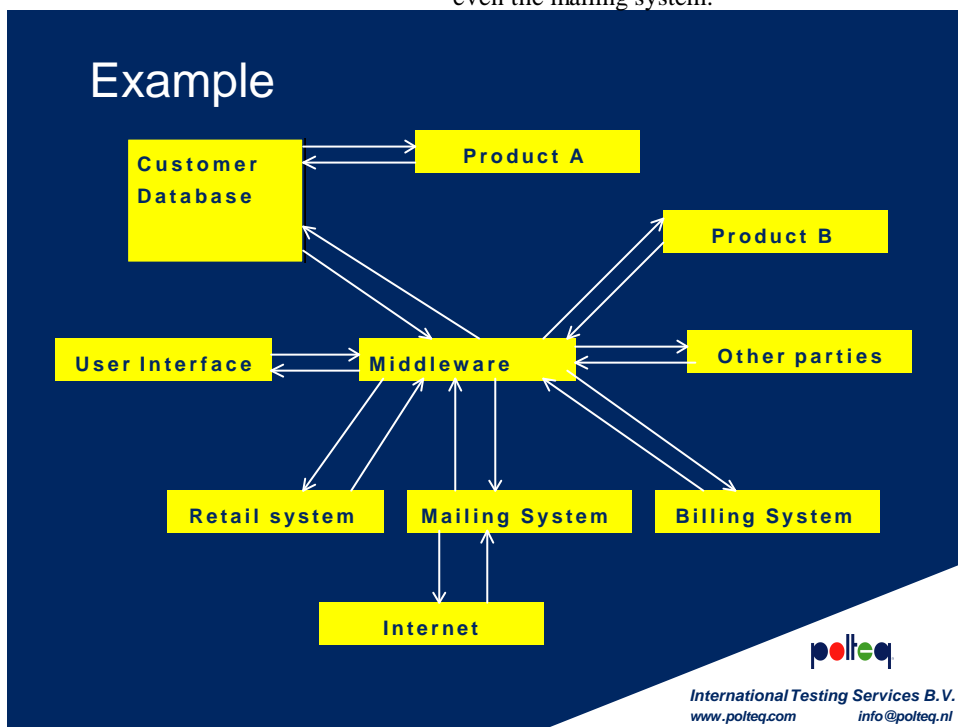
networks of antennas in the case of telecommunication. Several processes can be identified throughout the chain, each of them using various applications. Creating a customer for instance will make use of the user interface, the middleware and the customer database. Billing will be a process using product databases, the customer database, the middleware, the billing system and perhaps even the mailing system.

mean that regression testing is needed of all possible chains.

RISKS

Test projects, which are dealing with a chain of applications, face a lot of risks. These risks are related to the complexity of the project itself and the complexity of the product (the chain of applications itself). The risks especially relevant within a project involving a chain of applications are:

1. Organisational complications can arise because of the multitude of design, build and test teams and their



Afbeelding 3: Example of a chain

An example of a chain is given above. In this case a middleware application serves as the spider in the web of all connections between applications (with one exception). This model can be found in several areas of business, e.g. telecommunication, banking. The product application for instance, can be seen as administrative systems in the case of banking or physical

Projects will be dealing with these processes. Each project will affect parts of the overall system and might even have great impact on other projects. So: every project needs to assess the impact on other areas. This might imply possible regression tests having to take place on several other processes or parts of the system. For instance: if a, in itself, rather small project initiates a major re-building of the middleware this might

interdependencies as far as deliverables and delivery times are concerned. Very often it is believed that as long as everybody is producing his or her bit somehow, sometime, everything will fall into place at the end.

2. Documentation, coordinating the chain-functionality, is often lacking or not detailed or unambiguous enough. A chain test needs to be based on the integral design, information analysis and process

designs. These documents are often lacking, not detailed enough or unambiguous enough for the purpose of specifying the chain and therefore chain testing.

3. A misconception is in the field of interfaces. It is tempting to believe that as long as interface formats and procedures are coordinated and tested, the communication between applications will be covered and will work just fine. However, communication throughout the chain is not only determined by interfaces, also the intelligence in between the applications is of great importance. Therefore the processes and transactions that span the chain, require specific test scenarios to be developed, based on specifications for the chain functionality only. This kind of testing is often not well defined, based on the wrong techniques, started too late, not covered by the right tools and underestimated as far as the required capability of the testers is concerned.

I have been involved in a chain project for 3 years. The project was a brand new process, concerning the creation of transfer possibilities of facilities from one company to the other. Bases on the existing process and infrastructure a link and process had to be established between several mobile network operators in order for customers to be able to switch from one operator to another. This process was enforced by the Dutch government, with the intention to create more of a free market

interplay. The start date was fixed, enforced by heavy fining. The financial risk of the project was therefore high, increased even more by the fact that the mobile telephony market was booming in that period and lots of customers could be won and lost.

The functional test team was assigned to test the applications and links within the systems of 'our' company. The overall chain was to be tested at the end of the project by a small user group.

Within our systems 3 new applications were build, one of them being the middleware to be created and stand alone tested by a 3rd party.

The project implicitly had expected the suppliers to test their own application before delivery. These expectations were not met. Lots of defects were therefore found, especially in the middleware, which caused a lot of delay. This was partly mitigated by creating a large test team.

A lot of initially intended automation of processes was not achieved, partly because delays forced the project to cut functionality to be delivered, partly because a lot of manual processes were to be developed during build and test. These circumstances created a large mismatch between the expectancy of the customer and the business organisation on one hand and the delivered product on the other hand.

The project went live with a functionally solid product. In live the overall chain proved to be very instable. This caused a lot of further problems because during instability a lot of transactions got lost or were doubled.

Lots of fines had to be paid because of transactions not handled well. This caused lots

of legal procedures since none of the, competing, operators wanted to admit any responsibility. This also caused great delay in providing solutions.

After two years, during which some trouble shooting releases were done, it was decided to initiate a large project, aiming at automating a large part of the business process. This project never really got out of the design phase: after 5 months in which nothing real had been created due to a failing agreement process between the several teams and especially between the project and the business, the business stopped the project. A 3rd party, promising a quick and fancy interface to all applications, was involved. The software department of the company was only involved in maintenance of existing software. This project eventually completed, though with lots of unexpected costs, delay and infrastructure inconsistencies.

Achieving a successful chain test project

ORGANISATION / MANAGEMENT

Communication

Especially within complex projects, communication is vital.

Progress reports should be standardized and providing sufficient detail on activities to be completed. Progress reporting is of great importance, especially for testing since testing depends greatly on delivery time and order of delivery of project products. Progress reports should therefore be produced by all teams, on fixed dates (end of the week?) and distributed to all other teams. Regular project meetings

should be held from the start, where representation of every team is required. Strategies, plans, expectations and progress will be presented by each of these teams creating overall awareness of project progress. 3rd party suppliers should never lack at project meetings, for political and financial interests and conflicts might obscure the overall project coordination. Next to this a clear escalation process will have to be identified. If problems between teams occur it should be clear who has the authority to decide in these matters. Especially with 3rd party suppliers this is something that has to be negotiated in advance. If things go wrong in this field a stand still will easily occur (for instance when it comes to defect solving) and project progress is threatened.

Planning

The dependency on deliveries needs to be carefully mapped and responsibilities, activities and products should be clearly defined in detail. This means a close and detailed investigation in all dependencies, especially those on the critical path. Reporting on progress on each of the deliverables needs to be closely monitored and communicated centrally in the project meeting. There should be an integral, comprehensive, project plan which should be easily accessible to all and very accurate in actuality. Another good suggestion in order to keep the project manageable might be to divide the intended release into sub products. In this way basic deliverables can be proven early: the project will be more manageable because differences with the initial plan

will come in 'smaller chunks', the project strategy can be proven or adjusted in this way, buy-in can be achieved (and kept) from the organisation and the project group's motivation can be stimulated and maintained because of more easily achievable goals.

Quality and Risks

Testing normally includes a lot of time spend on activities which are not covering risks. There needs to be one or more thorough product quality risk assessments from which testing, but preferable also all other areas within the project, receives a priority list of areas and aspects to test. Product quality risks need to be identified and maintained throughout the project. For each one of them a strategy will be drawn up and progress on the coverage of these risks will be reported in the project meeting.

A complex project will almost inevitably experience delay because minor delays will be felt throughout the project. Time and money therefore require careful planning and efficiency will be a keyword. Quality and risk management actions will have to be carefully planned and divided amongst the project members. A firm test strategy, based on both product quality risks and project risks, will have to be implemented from a top down structure (from the chain to its parts). Each application testing will be fed from this central overall strategy and from risk assessments performed within the individual realm of the application only. In this way of doing so, expertise, existing within the entire project, will be used for a 'smart' test strategy.

Inspections

Reviews and inspections of all vital project documentation (including test documentation like plans and test cases) should be organized and performed as early and thorough as possible. Especially projects involving a chain will get benefit from structured reviews and inspections.

Testing the chain can only be done as the last part of the overall testing lifecycle. In a complex testing project the span of time might become too wide to let important defects be detected at the end. Mind you: severe defects might evoke a complete new lifecycle of testing. This means that inspections need to be planned from the beginning of the project. The people to perform these inspections need to have an overall knowledge of the chain, from both a business and a technical perspective. The problem with this will be the reference, the source for the inspections.

The documentation where the chain-specific specifications have been documented will have to be established. In general: the inspections will focus on the coherence of the global design with the individual documentation with special attention paid to processes spanning several applications.

The same argument will work for project documentation like project plans and test plans: by performing structured reviews 'defects' in these documents and therefore irregularities in the project work can be avoided. Also: an inspection process will make room for exchange of ideas and agreement on the project

planning and strategies.

Efforts put in inspections might seem to distract from other vital activities, but in fact prevent fault multiplication. Inspections and reviews will always pay off, not only as a prevention of severe defects, but also as a means to get agreement and knowledge!

Standardizing

In a chain not just the interfaces and communicating protocols need to be unified and agreed on, also the standards, definitions and templates used throughout the teams need to be clear to all within the project. Definitions of vital terms, especially for the completion phase of the project will have to be agreed on, or for the sake of time, be settled by the project manager. Especially severity classes of defects, support to be delivered and exit and entry criteria per phase in the overall lifecycle will have to be agreed and communicated in detail. A good suggestion is to produce a project set of definitions.

A project defined directory structure and templates (e.g. for progress reports and plans) will make communication, traceability and agreement more easy.

DOCUMENTATION

Specification

One of the misconceptions within chains is the thought that the overall coherence of the chain depends upon interfaces. Being able to process files or data from other applications doesn't mean the intended process will be fulfilled. There needs to be an overall co-ordinating specifying document that controls chain processing and which describes what kind of

information comes from where, will be processed by what and ends where. This document will be the basis for an additional level to the so-called V-model: the chain level.

Remember: testing is verifying if the product is in accordance with the specifications. So: if the system testing is verifying against the functional designs, who will test the overall coherence and based on which document? The chain test should verify the coherence and because of its importance should have its own mature specification document.

So, the traditional collection of documentation (global design, interface specifications and individual designs) is not sufficient to base a testing of the chain on. A global design will very often be too global and will only contain general descriptions of the chain functionality. The interface specifications will very often only indicate formats and processing of files. Individual designs will very often only describe processes and data up to the border with other applications. If none of these documentation levels describe the overall chain specifications, these need to be drawn up and communicated. Preferably the Global Design should provide a level of detail that could serve as a test basis for the chain testing. A chain test itself will not be based on the collection of all individual specifications per application but on an overall specification of the data communication (the global design or the so called information analysis). So, the problem for a chain test, in case no overall chain specification document is available, is deriving the specifications out of the existing documentation.

This means that very experienced and capable testers have to read through the documents, only focussing on data transfer between applications. Preferably all this should be maintained in one document, but this might be too much to ask for.

Archiving

A project is not a one time event. There will be successions, fixes etceteras. For this reason, but also for reasons of traceability, accountability and re-use in other projects, all test ware and project documentation will have to be archived in a rigid manner, supported by a firm version and configuration control. Mind you: there will be a lot of versions of applications, modules, documents and reports before going live. When this is not clearly archived, how can the organisation be sure what they are going to get out of a forest of versions?

Detail

Especially the test ware will need a high level of detail. The reason for this is that test cases will be highly complex to execute. Drawing up only logical test cases during the specification phase will be highly insufficient. What is needed is a clear description of data needed to be prepared in the various parts of the system before execution of the test case. This is a very underestimated aspect and should be covered by reserving capable resources in an early stage of the project.

CHAIN TESTING ASPECTS

Test Levels

A stand alone test of every application, interface testing followed by global coherence

testing (dataflow) is not sufficient. The problem with this is that data flow is testing with too many variables in each test case. In normal test projects this is no problem but if various parties are involved, analysing a defect with several variables spanning more than one application might become a political debate with no ending when defects occur which cannot be located in one of the parts of the chain. This will imply that a specialised and dedicated chain test team will have to be formed who will start working, parallel to all other project and test groups from the beginning of the project on. This test team will base their test cases on a risk assessment for the chain itself (or: the complete system). From these identified risks they will draw up a dedicated strategy, indicating which sub parts of the chains will be tested to which depth, using which techniques etceteras.

Testers

Chain testers will have to be able to get a quick insight into the overall chain so they can draw up their test cases. This is not just putting specialists from all applications in one room. Chain testers should be flexible, all-round testers with a quick learning faculty. Application specialists tend to be too slow and inflexible in incorporating knowledge from other systems. Another trait of a chain tester is stress immunity and an ability to stay focussed. Preparing, specifying, executing and completing a chain tests means to love frustration and communication under difficult circumstances and delay. This should not be underestimated.

Specification Techniques

The peculiarity of chain testing is to focus solely on transactions transcending one system and knowing how to write an overall case which covers the information going from one application to another.

This involves picturing all transactions and think of their coherence, both throughout the system as well as in time: can updates cross each other, thereby overwriting one another, make each other obsolete or disrupt processes and systems? Can data coming from one application create integrity or security breaks in another system? Have data been stored in too much places at the same time in the chain? How will transactions be taken further when the chain is instable?

Stand-alone functionality should not be tested within a chain test, although some regression testing in this field might be considered to be done in this phase of the project but this depends on results of previous phases.

So having extracted the relevant specifications for the chain tests, the chain itself might be divided into various processes, which in turn, depending on the risk assessment, might be covered by different depths of testing. This chain specification can be considered to be a system in itself and existing techniques can be used to derive test cases. Structured specification techniques are highly recommended because these offer:

- strategic manageability (focus on certain aspects and depth of testing),
- traceability (coverage level, reference to

documentation per test step),

- transferable test ware (when other resources or teams make use of the test scripts),
- more connectivity to automation tools.

Especially the Data Flow Specification technique will be fit for purpose although in some cases, say a data cycle test technique might be considered, i.e. when one application does a lot of creating, reading, updating and deleting in another application. Depending on the risk involved for this chain, boundary value analysis instead of equivalence partitioning might be chosen for a higher coverage level.

Simulation tools

Testing activities should be started as early as possible in the project, as we all now. Especially with chain testing this means that tools become even more important. Being able to test parts of the chain when the complete chain is not 'ready' yet implies the availability of tools, being capable of simulating missing parts of the chain. As early as possible an inventory should be made of what testable and meaningful subdivisions of the chain would be and what tools could be used to test these parts.

Record and play back tools can prove to be very helpful in regression testing. Regression testing will be relatively more important because various projects are more likely to use the same complex of applications. Other projects might need your regression scenario's because they will influence the technical structure of your processes! This will mean that the

maintenance of a complex test environment, maintenance of data sets and version control will be very important to make regression testing in this sense successful.

A business case should be drawn up, taking into account the costs for developing or purchase of the necessary tools and the benefits coming out of the use of the tool. This will mean a calculation of the number of releases in which such tools can be 'earned back'. Draw up a scenario with no tools and show what the consequence could be: project delay, inefficient use of resources, risks for going live etceteras.

testen van informatiesystemen en wordt afgesloten met het afnemen van het ISEB examen in de Engelse taal (multiple choice). Wij kunnen de opleiding zowel in de Nederlandse als in de Engelse taal verzorgen. In 2002 wordt de opleiding maandelijks gegeven; op verzoek kan deze ook in-house worden gegeven (overdag of 's avonds). Voor meer informatie: email info@polteq.com

Verschijningsdata TNN

De volgende nieuwsbrieven kunt u verwachten in:

- mei;
- augustus;
- november.

Ruim daar dus voldoende tijd voorin tijdens het werk.

Kortingsactie

We hebben verschillende kortingsacties die ons spontaan zijn aangeboden.

Als eerste een aanbieding van 'The Software Testing & Quality Engineering magazine STQE'. TestNet leden kunnen een abonnement met 35% korting krijgen. Hiervoor dient u het meegeleverde formulier te retourneren.

Ingezonden stuk

ISEB Foundation Certificate in Software Testing. Polteq is geaccrediteerd voor het opleiden van testpersoneel tot het ISEB Foundation Certificate in Software Testing. De opleiding geeft een algemene introductie tot het



Evenementen

Software Management and Applications of Software Measurement

PLAATS ANAHEIM CALIFORNIE
 GEBOUW DISNEYLAND HOTEL
 DATUM 11-15 FEBRUARI 2002
 TIJD
Informatie:
 URL: <http://www.sqe.com/smasm>

TestNet Thema avond

PLAATS AMERSFOORT
 GEBOUW DE EENHOORN
 DATUM 28 FEBRUARI 2002
 TIJD
Informatie: Onderwerp is usability testen voor webapplicaties

The 10th International Conference on Practical Software Quality Techniques

PLAATS NEW ORLEANS
 GEBOUW
 DATUM MARCH 4-8, 2002
 TIJD

The 4th International Conference on Practical Software Testing Techniques

PLAATS NEW ORLEANS
 GEBOUW HOTEL MONTELEONE
 DATUM MARCH 4-8, 2002
 TIJD
Informatie:
 URL: www.pstqconference.com

Quality Week

PLAATS BRUSSEL
 GEBOUW SHERATON
 DATUM 11 -15 MAART
 TIJD -
 URL: <http://www.qualityweek.com>
 Rita Bral Conference Director
qw@soft.com

TestNet Thema avond

PLAATS AMERSFOORT
 GEBOUW DE EENHOORN
 DATUM 24 APRIL 2002
 TIJD -
Informatie: Een special, nadere informatie volgt.

Algemene Ledenvergadering

PLAATS AMERSFOORT
 GEBOUW DE EENHOORN
 DATUM 30 MEI 2002
 TIJD -
Informatie: Nadere informatie volgt.

Voorjaarsfeest

PLAATS ZWOLLE
 GEBOUW BUITEN SOCIETEIT
 DATUM 13 JUNI 2002
 TIJD -
Informatie: Onderwerp: testen voor nieuwe ontwikkelomgevingen..

TestNet Thema avond

PLAATS AMERSFOORT
 GEBOUW DE EENHOORN
 DATUM 10 SEPTEMBER 2002
 TIJD
Informatie: Volgt nog.

Safecomp 2002

PLAATS CATANIA ITALIE
 GEBOUW
 DATUM 10 -13 SEPTEMBER 2002
 TIJD
Informatie:
 URL: <http://www.safecomp.org/>

Groot Najaarsfeest

PLAATS -
 GEBOUW -
 DATUM 31 OKTOBER 2002
 TIJD -
Informatie:

TestNet Thema avond

PLAATS AMERSFOORT
 GEBOUW DE EENHOORN
 DATUM 12 DECEMBER 2002
 TIJD
Informatie:.

Colofon

BESTUUR

Martin Pol	Voorzitter
Astrid Freericks	Penningmeester
Marco Jansen van Doorn	Secretaris Ledenadministratie
Hans van Loenhoud	2e voorzitter 2e penningmeester Marktverkenning, Informatievoorziening en Beheer
Bob van de Burgt	Marktverkenning, Informatievoorziening en Beheer
Rob van de Pouw Kraan	Internet
Gerrit de Munck	Communicatie & Evenementen
Frank van Elsdingen	

COMMUNICATIE & EVENEMENTEN

Frank van Elsdingen
 Egbert Egberts
 Rob Hendriks
 Milo van der Kruis
 Gerrit de Munck
 Meile Posthuma
 Fred Weber
 E-mail: cie-ce@testnet.org (algemeen)
 E-mail: evenementen@testnet.org (aanmelden)

LID WORDEN

U kunt lid worden door een e-mail te sturen naar de ledenadministratie of door op onze internet site het online registratieformulier in te vullen.
 Internet site: www.testnet.org

LEDENADMINISTRATIE

Marco Jansen van Doorn
 E-mail: ledenadministratie@testnet.org

REDACTIE

Milo van der Kruis
 Meile Posthuma
 E-mail: tnn@testnet.org

TESTNET NIEUWS[©]

TestNet Nieuws verschijnt eenmaal per kwartaal. Kopij aanleveren per e-mail van de redactie
 Het is niet toegestaan om de nieuwsbrief of delen eruit zonder bronvermelding over te nemen.



Your information source for building better software!
 TestNet Organisation Members
SAVE 35%!
 Courtesy of Software Quality Engineering

Start my special rate subscription to STQE magazine!
 1 year (six issues) £39* 2 years (twelve issues) £78

First Name: _____ Surname: _____
 Title: _____ Organisation: _____
 Telephone: _____ Fax: _____
 Email: _____
 Street Address: _____
 City: _____ State/Province: _____
 Zip/Postal Code: _____ Country: _____
 VISA/MasterCard/Eurocard/Delta #: _____ Expiry Date: _____
 Printed Name on Card: _____
 Signature (Required): _____
 Cheque Enclosed payable to SQE Europe: \$ _____

Job Function (Please Pick One):

<input type="checkbox"/> App. Architect	<input type="checkbox"/> Developer	<input type="checkbox"/> Project Leader	<input type="checkbox"/> SW Process Eng.	<input type="checkbox"/> Test Manager
<input type="checkbox"/> Business Analyst	<input type="checkbox"/> Development Mngr	<input type="checkbox"/> QA Analyst	<input type="checkbox"/> Systems Analyst	<input type="checkbox"/> Training Coordinator
<input type="checkbox"/> CIO	<input type="checkbox"/> Info. Tech. Mngr	<input type="checkbox"/> QA Manager	<input type="checkbox"/> Technologist	<input type="checkbox"/> User
<input type="checkbox"/> Consultant	<input type="checkbox"/> Meas. Analyst	<input type="checkbox"/> Software Eng.	<input type="checkbox"/> Test Engineer	
<input type="checkbox"/> CTO	<input type="checkbox"/> Programmer	<input type="checkbox"/> Software Mngr.	<input type="checkbox"/> Test Lead	

Industry (Please Pick One):

<input type="checkbox"/> Computer-Related Manufacturing	<input type="checkbox"/> Health Care/Medical	<input type="checkbox"/> Tele/Data Communications
<input type="checkbox"/> Consulting/Training	<input type="checkbox"/> Manufacturing (non-computer)	<input type="checkbox"/> Transportation/Utilities
<input type="checkbox"/> Financial Services/Banking/Insurance	<input type="checkbox"/> Outsource Testing/QA/Development	<input type="checkbox"/> Web/E-business
<input type="checkbox"/> Government/Military/Aerospace	<input type="checkbox"/> Retail/Commercial Software	<input type="checkbox"/> Other: _____

No-Risk Offer:
 If you are not satisfied with your first issue of STQE, simply write "cancel" on the subsequent invoice/statement and return it within 30 days for a full refund.

SIMPLY FILL OUT THIS FORM AND SEND IT BACK TO US AT:
 FAX: +44 (0) 1702 290173
 SQE Europe, P.O. Box 2046, Shoeburyness, Essex SS3 9UA
 Phone: +44 (0) 1702 295796
www.stqemagazine.com

PRIORITY CODE: EPZ235TN

TestNet Nieuws

