

SOx IT Testing process compliancy for Shell People



**Afstudeerscriptie Master of Science Software Engineering
Universiteit van Amsterdam, Hogeschool van Amsterdam en de Vrije Universiteit.**

Auteur: Ing. Onno Bagijn
Begeleider: Drs. Rolf van den Doel (Shell People Programme)
Afstudeerdocent: Prof. Dr. Paul Klint (UvA, CWI)
Opdrachtgever: Shell International B.V.
Periode: 4 April tot 1 Juli 2005
Datum: Vrijdag 12 augustus 2005

Inhoudsopgave

1. SAMENVATTING	2
2. INTRODUCTIE.....	3
2.1. PROBLEEMSTELLING/ONDERZOEKSVRAAG	3
2.2. PLAN VAN AANPAK	4
2.2.1. <i>Activiteiten</i>	4
2.2.2. <i>Verwachte resultaten</i>	5
2.3. SARBANES-OXLEY ACT	5
2.4. SOFTWARE ONTWIKKELING EN ONDERHOUD BIJ SHELL PEOPLE PROGRAMME.....	6
3. TESTTRAJECT	9
3.1. DEFINIËREN VAN DE TESTSTRATEGIE.....	9
3.2. KNELPUNTEN.....	11
3.3. VERBETERING VAN DE SOFTWARE TESTPROCESSEN	11
3.3.1. <i>Teststrategie (Standaardisatie/harmonisatie)</i>	11
3.3.2. <i>Hergebruik van test scripts</i>	12
3.3.3. <i>Risicoanalyse</i>	14
3.3.4. <i>Test Data Hulpmiddel</i>	16
3.3.5. <i>Software inspectie</i>	16
3.3.6. <i>Testplan</i>	18
3.3.7. <i>Formele goedkeuring</i>	18
4. INTEGRATIE SOFTWARE ONTWIKKELHULPMIDDELEN.....	19
5. RESULTATEN/CONCLUSIE.....	20
6. EVALUATIE.....	21
7. LITERATUUR.....	22
8. BIJLAGEN.....	24

1. Samenvatting

Deze afstudeeropdracht is uitgevoerd bij de Personeelsafdeling (Shell People) van Shell International B.V. en gaat over verbetering van de testmethodiek voor het SAP-salarissysteem vereist door SOx (Sarbanes and Oxley Act).

SOx wetgeving is bedoeld om de integriteit van jaarverslaggeving te waarborgen. Het is ingevoerd in 2004 in de Verenigde Staten naar aanleiding van de financiële malversaties bij Enron en Worldcom. Shell, genoteerd aan de New York Stock Exchange, dient vanaf 1 januari 2007 aan deze wetgeving te voldoen.

Begin 2004 bleek dat Shell de bewezen olie- en gasreserves vermeld in het jaarverslag over 2003 te hoog had opgenomen. Dit leidde tot veel negatieve publiciteit over Shell en claims van aandeelhouders en andere belanghebbenden. Zo werd een schikking getroffen met de Security and Exchange Commission (waakhond van de Amerikaanse Beurs) voor een bedrag van 117 milj. US dollar. Shell is er daarom op gebrand aan de SOx-richtlijnen te voldoen en zijn reputatie als solide onderneming te herstellen.

Uit een al eerder in het kader van SOx gehouden vooronderzoek kwam de noodzaak voor een duidelijke teststrategie voor de gehele organisatie naar voren.

Een eerste stap voor verbetering van het testproces was daarom specificatie en goedkeuring van een algemeen toepasbare teststrategie die voor alle testteams tot harmonisatie van het testen leidt.

Vervolgens zijn de volgende knelpunten geïdentificeerd:

- Testscripts worden onvoldoende hergebruikt.
 - o Testscripts worden niet goed beheerd.
- De diepgang van testen wordt subjectief bepaald.
- De betrouwbaarheid van gebruikte testdata laat te wensen over.
- Controle op kwaadwillende code wordt onvoldoende gericht uitgevoerd.
- Testplannen voor systeemwijzingen worden vaak niet opgesteld.
- Formele goedkeuring van het testen en de registratie daarvan vinden niet altijd plaats.
- Opslag van SOx-relevante documenten in systemen als Livelink (document management system) en Greenlight (SOx controle beheersysteem) vergt veel zoektijd voor verificatie- en afstemmingsdoeleinden.

Oplossing van de voorgaande knelpunten heeft geleid tot de volgende verbetervoorstellen/-acties:

- Het hergebruik van test scripts wordt sterk vergemakkelijkt, aangetoond middels een door mij ontwikkelde demo van een testhulpmiddel.
 - o Gestructureerder testen door toepassing van het door mij ontwikkelde demosysteem resulterend in een efficiënter beheer van test scripts.
- Het testen kan objectiever en gericht gebeuren door de invoering van een algemeen geaccepteerd risicoanalyse systeem, waarvoor ik een risico matrix heb ontwikkeld.
- Gebruik van een testhulpmiddel voor het creëren van betrouwbare testdata.
- Ontwikkeling van een controlelijst voor software inspectie, waaronder inspectie van kwaadwillende code.
- Het verplicht opstellen van testplannen voor projecten van geringe omvang en voor relevante systeemwijzingen.
- Aanscherpen van de regels voor formele goedkeuring en registratie van testen.
- Integratie van Livelink en Greenlight. Helaas is dit laatste niet mogelijk omdat Greenlight, een pakket van derden, geen communicatie toestaat.

De teststrategie is op 31 mei 2005 goedgekeurd door het Shell People Leadership Team en is nu officieel de teststrategie voor Shell People Programme geworden.

Door implementatie van bovenstaande oplossingen is het testen binnen Shell People zodanig verbeterd dat het voldoet aan de SOx eisen met betrekking tot applicatie controles.

Bovendien worden zowel in de materiele als immateriële sfeer baten behaald zoals een tijdsbesparing in het testen van 0.5 man per jaar en vermindering van het aantal fouten in de software.

2. Introductie

In dit verslag wordt vaak verwezen naar SP en SPP. Met SP (Shell People) wordt bedoeld de afdeling Personeelszaken binnen Shell terwijl SPP (Shell People Programme) de automatiseringsorganisatie betreft binnen SP. Deze organisatie is wereldwijd verantwoordelijk voor ontwikkeling en implementatie van het SAP-HR (Human Resource) systeem.

Afstudeeropdracht

Als afsluiting van de studie Master of Science Software Engineering is de afstudeeropdracht in overleg met Shell International B.V. bij het Shell People Programme (SPP) uitgevoerd.

De afstudeeropdracht gaat over:

Het verbeteren van de testmethodiek in het Shell People Programme zodanig dat aan de SOx Algemene IT Controle eisen wordt voldaan.

De onderzoeksvraag is een gevolg van de recente nieuwe eisen die onder andere aan de ontwikkeling en het onderhoud van softwarepakketten is gesteld binnen Shell door de SOx-wetgeving.

Sarbanes Oxley

Als gevolg van de financiële debacles van de Amerikaanse maatschappijen ENRON en Worldcom is in 2002 de Sarbanes Oxley wetgeving (SOx) in de V.S. van kracht geworden om daarmee bezitters en mogelijke kopers van aandelen die op de Amerikaanse beurs zijn genoteerd te beschermen tegen onjuiste financiële verslaggeving. Shell, genoteerd aan de New Yorkse aandelenbeurs, moet met ingang van 1 januari 2007 aan de SOx-wetgeving voldoen.

Shell People Programme

Het Shell People Programme (SPP) is een 5-jarig programma dat als doelstelling heeft het wereldwijd harmoniseren en standaardiseren van de Human Resource (HR) processen binnen Shell, ondersteund door één globaal SAP R3 systeem.

Het SAP R3 HR systeem ondersteunt o.a. salarisadministratie functionaliteit voor diverse Shell bedrijven in meer dan 40 landen. Het HR systeem is een belangrijk element in de financiële rapportage binnen de Shell Groep en moet aan de SOx controle-eisen voldoen.

In paragraaf 2.1 “Probleemstelling/Onderzoeksvraag” zal de onderzoeksvraag beschreven worden. In paragraaf 2.2 wordt het plan van aanpak gepresenteerd terwijl in 2.3 het effect van de SOx wetgeving en van de SOx Shell richtlijnen op het software ontwikkelproces wordt uitgelegd.

Voor de toepasbaarheid van verschillende technieken uit de wetenschappelijke literatuur past SPP een specifiek ERP onderhoud- en ontwikkelmodel toe. Deze komt niet volledig overeen met de traditionele ontwikkel- methoden. Derhalve worden in paragraaf 2.4 de bevindingen van het SPP ERP maintenance/implement model weergegeven.

2.1. Probleemstelling/Onderzoeksvraag

Bedrijfsprocessen die financiële rapportages ondersteunen zijn sterk afhankelijk van IT systemen en de controles die geïntegreerd zijn in zowel de applicatie als de onderliggende IT infrastructuur. Shell heeft zogenaamde controleregisters opgesteld samen met externe accountants. Deze Controle Registers beschrijven waaraan de financiële bedrijfsprocessen (b.v. C9 Payroll & Benefits Controls Register), de IT infrastructuur (C12 Controls Register) en Applicatie Onderhoud/Beheer (C11 Controls Register) moeten voldoen. Voor SPP, als leverancier van HR applicatiediensten, is specifiek het C11 Controle Register van belang.

Door een voorafgaand onderzoek begin 2005 naar de mate waarin SPP voldeed aan de eisen van het C11 register is gebleken dat o.a. op het gebied van testen nog niet volledig wordt voldaan aan de C11 eisen. Er was bijvoorbeeld geen volledig gedocumenteerde teststrategie.

Onderzoeksvraag

Onderzoek de doeltreffendheid en doelmatigheid van de voor geheel SPP van toepassing zijnde testmethodiek.

Daartoe delen we het onderzoek op in 3 delen:

- Definieer de teststrategie voor Shell People.
- Analyseer de huidige manier van testen en identificeer knelpunten die bestaan.
- Specificeer verbeteringen in het testproces.

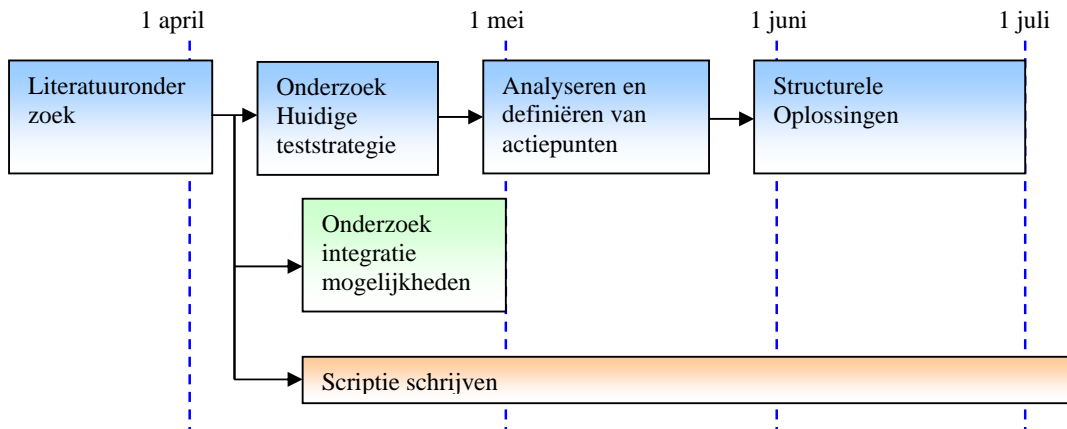
Naast definiëring van een teststrategie zal ik me bovendien bezig houden met een onderzoek naar:

- Welke mogelijkheden er bestaan voor automatische integratie van de SPP SDLC (Software Development LifeCycle) en ondersteunende hulpmiddelen als Global Livelink (knowledge management system) en Greenlight. (SOx control tool)
 - Bestudeer de huidige hulpmiddelen.
 - Onderzoek de mogelijkheden van integratie van de verschillende hulpmiddelen.
 - Specificeer een Integratie plan.

2.2. Plan van aanpak

2.2.1. Activiteiten

De uit te voeren activiteiten om de verwachte resultaten van het onderzoek te bereiken zijn opgedeeld in 4 fasen.



Voor het onderzoek naar de teststrategie zullen de volgende fasen doorlopen worden:

Literatuuronderzoek

Wat houdt SOx in?

Bestudering van gestructureerde, doeltreffende testmethoden/technieken

Welke technieken/methodes worden er gepropageerd?

Onderzoek huidige teststrategie

Hoe wordt er nu getest? Wat is de werkwijze voor het testen? Wie zijn verantwoordelijk voor de kwaliteit van een wijziging?

Tijdens deze fase worden er vergaderingen/interviews gehouden met verschillende verantwoordelijke medewerkers van betrokken afdelingen om op een gestructureerde wijze de huidige teststrategie te onderzoeken.

Analyseren en definiëren van actiepunten

Waar liggen de knelpunten in de huidige teststrategie om aan de SOx eisen te voldoen?

Structurele Oplossingen

De oplossingen die voortaan uitgevoerd moeten worden om efficiënter en doeltreffender te testen en tevens aan de SOx eisen te voldoen.

Voor het onderzoek naar de mogelijkheid voor integratie van Global Livelink en Greenlight met de Software Development LifeCycle zijn de volgende activiteiten uitgevoerd:

- Onderzoek naar de mogelijkheden voor integratie van de bovengenoemde software hulpmiddelen.
- Het experimenteren om aldus de mogelijkheden voor integratie uit te proberen.

2.2.2. Verwachte resultaten

De volgende producten worden als resultaat van de onderzoeksopdracht opgeleverd:

- Scriptie
- SPP teststrategie
Een teststrategie die voor geheel SPP geldt en die voldoet aan de SOx eisen.
- Lijst met bevindingen/Actiepunten
Opstellen van de knelpunten die zijn gevonden tijdens het onderzoeken van de huidige situatie.
- Voorstellen om de gesignaleerde knelpunten op te lossen.
Omschrijving van de oplossingen voor het verhelpen van de knelpunten. Tijdens het onderzoek zal ik me vooral richten op de volgende knelpunten:
 - Risicomatrix
 - Software inspectie
 - Hergebruik (geautomatiseerde) test scripts

2.3. Sarbanes-Oxley act

Zoals in de Introductie aangegeven, beschrijft SOx regels om bedrijven te dwingen een verantwoordelijk bestuur te voeren. De integriteit, betrouwbaarheid en beveiliging van de informatie binnen het bedrijf is zeer belangrijk [1]. Immers, vandaag de dag wordt bijna elk bedrijfsproces op de een of andere manier ondersteund door informatiesystemen. Derhalve is het voor een bedrijf van groot belang dat de informatie opgeslagen in deze systemen goed beveiligd is en integer is en tot rapportering van betrouwbare informatie leidt.

Zoals in de vorige paragraaf aangegeven heeft Shell een reeks van controleregisters ontwikkeld voor het beheersen van SOx-relevante bedrijfsprocessen waaronder C11. Dit register controleert de ontwikkeling en het onderhoud van software applicaties. De registers zijn samen met Ernst & Young opgesteld en zijn voor een deel gebaseerd op CoBiT (Control Objectives for Information and related Technology). CoBiT is een algemeen geaccepteerde standaard voor opzet van een controleraamwerk voor beheersing van Informatie Technologie.

C11 bestaat uit de volgende controlegebieden:

1. Toepassing van een Software Ontwikkeling Methodologie.
2. Hantering van een Systeemwijzigingsprocedure.
3. Beheer van Noodsysteemwijzigingen.
4. Onderzoek en goedkeuring Systeemontwerp.
5. Het opstellen van een Applicatietestplan.
6. Het testen van systeemwijzigingen.
7. Acceptatietest van projecten/systeemwijzigingen.
8. Promotie van projecten/systeemwijzigingen van de Test – naar de Productieomgeving.
9. Organisatie van Dataconversie.
10. Opzet voor een raamwerk voor dienstverlening door derden of door interne dienstverleners.
11. Identificatie, Authenticiteit en toegang tot applicaties.
12. Beheer gebruikerstoegang tot applicaties.
13. Beheer van systeemproblemen en escalatie er van.
14. Back-up en herstel procedures.
15. Data authenticiteit en integriteit van data.

Teneinde aan de SOx-eisen te voldoen heeft Shell een methodologie ontwikkeld die voor C11 onder meer de volgende onderwerpen bevat:

- Analyse van het huidige proces door het invullen van de controlegebieden van het C11 register waarbij de actuele controles worden beschreven en worden vergeleken met de SOx controlerichtlijnen waarna waar nodig verbeteringen worden gedefinieerd.
- Het opstellen van een plan voor uitvoering van de verbeteringen.

- Het onderwerpen van de Controleregisters aan een kwaliteitscontrole door een onafhankelijk team.
- Na uitvoering van de verbeteringen, het opstellen van test scripts om na te gaan of het C11 proces nu aan de SOx-normen voldoet.
- Het testen van de controles middels de gedefinieerde test scripts tot alle controleposten aan de SOx-regels voldoen.
- Het uitvoeren van een onafhankelijk test door Internal Audit om te zien of werkelijk aan SOx-regels wordt voldaan.
- Aanpassing van de controleregisters met het van Internal Audit ontvangen commentaar.
- Uitvoering van een controle door Externe Accountants die tot verklaring van SOx-compliance voor het C11 proces moet leiden na verwerking van het commentaar ontvangen van Externe Accountants.

De genoemde methodologie geeft aanwijzingen voor uitvoering van de hiervoor genoemde stappen zoals bijv.;

- Welke stappen dienen bij ontwikkeling en onderhoud van systemen gehanteerd te worden?
- Welke functiescheidingen dienen hierbij te worden gehanteerd?
- Hoe dient elke stap door betrokken partijen te worden afgetekend?
- Welke documentatie dient als bewijs te worden bewaard?
- Hoe groot dient de steekproefomvang te zijn voor het uitvoeren van testen en hoe dient te worden getest?

De problematiek rond de te hoge rapportering van olie- en gasreserves en als gevolg hiervan de deuk in het imago van Shell heeft er toe geleid dat Shell een zeer groot belang hecht aan het SOx-compliant maken van zijn bedrijfsprocessen om daarmee het vertrouwen van de financiële markt in Shell terug te winnen.

SOx versus ISO (International Organization for Standardization)

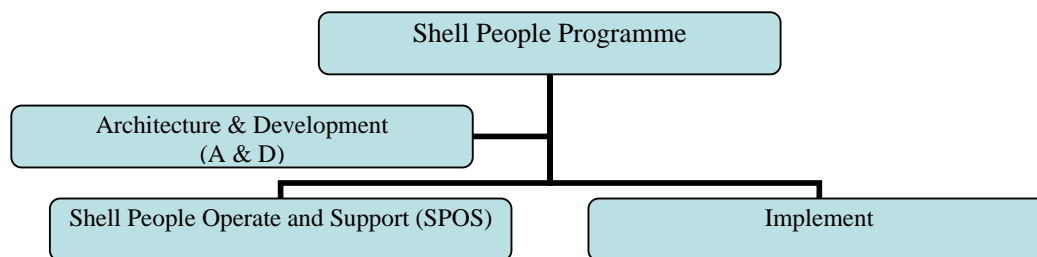
Binnen Shell bestaat al jaren veel aandacht voor het verkrijgen van ISO-certificaten op velerlei gebied waaronder de ontwikkeling en het onderhoud van systemen.

SOx- en ISO standaarden verschillen omdat SOx zich richt op de betrouwbaarheid van de financiële verantwoording van in de VS op de Beurs genoteerde ondernemingen waarbij een jaarlijkse toetsing plaatsvindt of aan de gestelde controle eisen is voldaan.

ISO daarentegen richt zich op het certificeren van projecten en bedrijfsprocessen.

Kwaliteitstandaarden/procesverbeteringsmodellen zoals ISO 9001, SPICE, CMMI beschrijven kwaliteitsnormen en zorgen dat er een juiste software kwaliteitsbewaking plaatsvindt om verantwoord het ontwikkelproces te laten verlopen. Men richt zich met deze standaarden op het proces omdat men veronderstelt dat een goed georganiseerd proces tot een kwalitatief beter product leidt [2].

2.4. Software ontwikkeling en onderhoud bij Shell People Programme



Shell People Programme (SPP) is wereldwijd verantwoordelijk voor de IT-systemen van Shell People Services (human resource and payroll services globally). SPP voert een 5-jarig programma uit in 40 landen om tot harmonisatie te komen van alle personeelssystemen. SAP HR (Human Resources) dient dan in één universele versie in alle betrokken landen gebruikt te worden.

Het belangrijkste systeem is het salarisadministratiesysteem, dat gebruik maakt van het SAP-pakket. Op dit moment is SAP HR reeds in 30 landen geïmplementeerd. Het projectbudget voor invoering van SAP HR in 40 landen bedraagt 360 miljoen US dollar.

A&D is verantwoordelijk voor ontwikkeling van middel tot grote projecten en ondersteunt waar nodig de andere afdelingen.

SPOS is verantwoordelijk voor het onderhoud en de ondersteuning van de geïmplementeerde SAP systemen.

De afdeling Implement is verantwoordelijk om SAP te implementeren in de verschillende landen.

Grote organisaties zoals Shell, Philips, Unilever, Heineken etc. gebruiken allemaal ERP pakketten zoals SAP om bedrijfsprocessen op geautomatiseerde wijze te ondersteunen. Het implementeren en onderhouden van zulke informatiesystemen voor ondersteuning van bedrijfsprocessen is in de loop der tijd veranderd.

Tot 1995 vond binnen Shell veelal ontwikkeling van maatwerksystemen plaats. In 1995 werd besloten om het SAP pakket waar mogelijk als geïntegreerde standaardoplossing wereldwijd te gaan toepassen. Aanvankelijk vond implementatie van ERP-pakketten per land plaats en werd het ERP-pakket relatief veel aangepast. Dit bleek een kostbare oplossing en stelde niet in staat om aan de steeds verdergaande wens tot harmonisatie van bedrijfsprocessen op wereldniveau te komen.

Sinds een aantal jaren vindt ontwikkeling plaats van standaardoplossingen op regioniveau (een of meer werelddelen) of wereldniveau waarbij het aantal aanpassingen zoveel mogelijk wordt beperkt.

Het bovenstaande heeft er toe geleid dat voor bedrijfsapplicaties bijna geen programmacode meer wordt ontwikkeld. Wanneer programmering moet plaats vinden wordt dit uitbesteed en dan voornamelijk in landen als Maleisië en India.

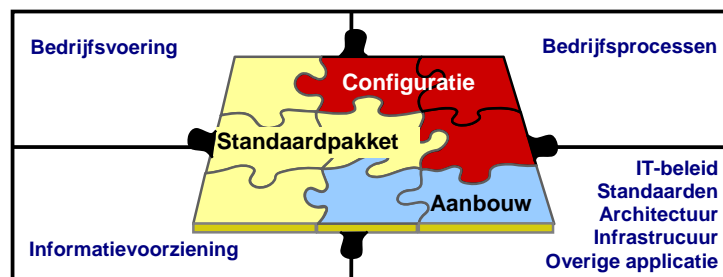
Het beheer van het SAP-pakket vindt centraal plaats waarbij het functioneel (logisch ontwerp) beheer door Shell zelf gebeurt terwijl het technisch (programmering en infrastructuur) beheer veelal wordt uitbesteed.

Het toepassen en onderhouden van ERP pakketten wijkt af van het traditioneel ontwikkelen van software [3, 4], maar brengt niet minder risico met zich mee [5].

Een belangrijk verschil is dat er vooral softwareconfiguratie plaatsvindt en het programmeren met behulp van bijvoorbeeld ABAP (programmeertaal voor SAP systemen) wordt uitbesteed of door een kleine groep intern wordt uitgevoerd. De programmatuur van het ERP systeem (zoals de C++ programmatuur van het SAP R/3 systeem) is de verantwoordelijkheid van de leverancier. Het behouden van de integriteit van de uitgebreide functionaliteit en samenhang ervan met andere modules levert veelal hoofdbrekens op.

Zoals eerder opgemerkt streeft Shell uit kosten- en beheersbaarheidsoverwegingen ernaar om aanpassingen aan het SAP-systeem zoveel mogelijk te beperken.

In-huis ontwikkeling zorgt voor een passend systeem bij de bedrijfsprocessen [6, 20], terwijl bij het toepassen van een ERP systeem bedrijfsprocessen ingepast dienen te worden in het systeem zoals is weergegeven in afbeelding 1.



Afbeelding 1 – ERP pakket passend in de organisatie. [8]

Bij een ERP-pakket als SAP moet de keuze gemaakt worden tussen het aanpassen van bedrijfsprocessen aan de door SAP gehanteerde/voorgestane bedrijfsprocessen of het aanpassen van SAP aan de specifieke bedrijfsprocessen. Het laatste levert het risico op dat de ontwikkeling en het onderhoud van het pakket kostbaar zijn en dus veel inspanning vergen, met name bij invoering van nieuwe versies van het pakket.

Kosten voor zelfgemaakte onderdelen zullen het meeste onderhoud vragen.

Het testen is gericht op het vaststellen van het correct functioneren van de programmatuur en de ondersteuning van bedrijfsprocessen.

Kwaliteitsattributen zoals gebruikersvriendelijkheid, hergebruik, installeerbaarheid en andere niet-functionele eisen zijn hoofdzakelijk de verantwoordelijkheid van de leveranciers van het ERP pakket. Inpasbaarheid, functionaliteit, prestatieniveau, het oppakken en begrijpen (opleiding, veranderingsbegeleiding, ondersteuning etc.) van functionaliteit zijn vooral van belang voor eindgebruikers van het ERP systeem en dus de verantwoordelijkheid van SPP.

Beide hiervoor geciteerde artikelen [3] en [4] geven aan dat er weinig wetenschappelijke artikelen zijn gepubliceerd die ingaan op de nieuwe problematiek die voortkomt uit het implementeren en onderhouden van ERP pakketten.

Bij SP wordt er hoofdzakelijk gebruik gemaakt van SAP. Dit vereist een ander paradigma inzake software onderhoud/ontwikkeling dan bij zelfontwikkelde systemen. Standaard methodieken en technieken voor ERP-systemen worden toegepast waarbij rekening wordt gehouden met de specifieke problematiek die voor SP geldt.

Het SP-systeem is omvangrijk en complex omdat het bedrijfsproces dat wordt ondersteund complex is en zeer veel landen met veelal een afwijkend belastingregime bediend.

Het beheren van de informatiesystemen blijft vooral mensenwerk. Het constant evalueren en onderhouden van bedrijfsprocessen moet er voor zorgen dat het systeem de bedrijfsprocessen correct blijft ondersteunen.

Als gevolg van een steeds sneller veranderende wereld dienen bedrijfsprocessen in de loop der tijd, wanneer nodig, aangepast te kunnen worden.

De zwakte van ERP-pakketten is dat aanpassing van de in de standaard software opgenomen bedrijfsprocessen niet eenvoudig is en traag verloopt.

3. Testtraject

Dit hoofdstuk beschrijft de aanpak en problematiek rond het testen binnen SP.

Testen is meestal een onderschatte en niet populaire activiteit, maar is een belangrijke en onmisbare activiteit bij ontwikkeling en onderhoud van software. Testen is nodig om inzicht te krijgen in de kwaliteit van het systeem en de risico's bij het in productie nemen. Testen is daarom nog steeds de primaire manier om de kwaliteit van het systeem vast te stellen.

De teststrategie vormt de basis voor een gestructureerd en beheersbaar testtraject [8]. Om te voldoen aan de SOx controle-eisen is het belangrijk om een beeld van de testactiviteiten te krijgen.

Ten behoeve van de formulering van een teststrategie is eerst onderzocht wat de globale activiteiten zijn en hoe deze qua testen worden uitgevoerd. In paragraaf 3.1 is beschreven hoe de teststrategie globaal is opgezet. In paragraaf 3.2 zijn de knelpunten aangegeven tussen de huidige situatie en de nieuwe teststrategie terwijl in paragraaf 3.3 de oplossingen worden gepresenteerd die de testactiviteiten op een doeltreffender wijze in lijn brengen met de gedefinieerde teststrategie.

Het belang van testen is zeer groot omdat aan een salarisadministratiesysteem hoge betrouwbaarheidseisen worden gesteld.

Voor de gehele implementatie van SP-systemen in verschillende landen is van het totale budget 13,3 procent besteed aan testen. Dit is exclusief de kosten die door eindgebruikers voor het uitvoeren van acceptatietesten zijn gemaakt.

Volgens een beperkt onderzoek bij SPP uit 2003 en uit gegevens van recent afgeronde projecten blijkt dat rond 20 procent van de totale kosten van een SAP-implementatieproject wordt besteed aan testen. Een onderzoek uit 1995 concludeert dat er over het algemeen per software ontwikkelingsproject (eigen ontwikkeling van programmatuur) 30 t/m 50 procent besteed wordt aan testen [21].

Volgens het National Institute of Standards & Technology [7] gaat er per jaar in de VS, 22.2 tot 59.5 miljard dollar verloren als gevolg van softwarefouten voortkomend uit hantering van inadequate testfaciliteiten. Een rapport van het U.S. Department of Defense geeft aan dat alleen al door software fouten jaarlijks vier miljard dollar wordt verloren [7] in hun organisatie.

Ondanks het gebruik van ERP-pakketten wordt er relatief veel tijd aan testen besteed.

3.1. Definiëren van de teststrategie

Voor het onderzoeken van de doeltreffendheid en de doelmatigheid van het testen door SPP zijn eerst de testactiviteiten voor geheel SPP in kaart gebracht en geanalyseerd om vervolgens met eventuele verbeterpunten te kunnen komen.

Om te beginnen is er onderzocht welke activiteiten en testen er plaatsvinden bij SPP. De eerste fase is het categoriseren van de hoofdactiviteiten van SPP:

Project	Een project is over het algemeen een specifieke landenimplementatie, uitgevoerd door de afd. 'Implement', maar kan ook grote implementaties betekenen van nieuwe (wereldwijde) toepassingen zoals Manager Self Service (applicatie voor beheer van personeelsgegevens) uitgevoerd door de A&D afdeling.
Breakfix, maintenance	De functionaliteit die naar behoren werkte en gerepareerd moet worden op de SAP productieomgeving, wordt een 'breakfix' genoemd. De prioriteit is niet dringend daarom worden deze wijzigingen toegevoegd aan de wekelijkse bundel met wijzigingen. De functionaliteit die nieuw is, maar geen ontwerp- of procesveranderingen omvatten wordt 'Maintenance'-wijziging genoemd. Dit zijn wijzigingen zoals nieuwe loontypes, werkschema's, vakantie kalenders enz. Deze veranderingen zullen worden behandeld op dezelfde manier als een Breakfix.
Enhancements	Nieuwe functionaliteit die wordt toegevoegd aan het systeem. Uitgevoerd door de SPOS afdeling.
Emergency	Break-fixes die een grote invloed hebben op essentiële processen en die de betaling aan medewerkers beïnvloeden, waarvoor geen geldig 'workaround' beschikbaar is, worden behandeld als 'Emergency'. De wijziging wordt in dit geval direct in productie geplaatst.
HRSP	Human Resources Support Packages (HRSP). HRSP's zijn wijzigingen die vereist zijn als gevolg van veranderende bedrijfsbehoeften (bijv. wettelijke veranderingen) en voor de correctie van bestaande fouten. HRSP wordt middels versiebeheer drie keer per jaar toegepast (April, September en December).

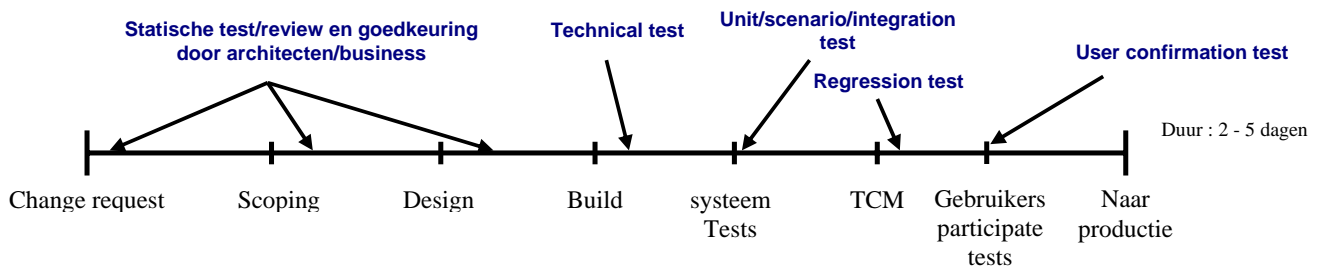


Small Project	'Small projects' worden vooral uitgevoerd door A&D (e.g. archiveringsproject, lijst met onderhoudsverzoeken, ITS 6.2 upgrade). De geringe (qua omvang en complexiteit) projecten vereisen niet het toepassen van de gehele ontwikkelprocedure/methodiek maar een gedeelte van de methodiek.
Annual Merit Review	Herweging van het pensioenbasissalaris voor lokale medewerkers en voor medewerkers die in het buitenland werken.
Interfaces (WBI/iHub).	Interfaces ontwikkeld door middel van iHub/IBM Websphere Business Integrator (WBI) product. Interface ontwikkeling wordt uitgevoerd als onderdeel van 'projects', 'enhancements' op de geïnstalleerde systemen, 'breakfixes' en 'maintenance'.

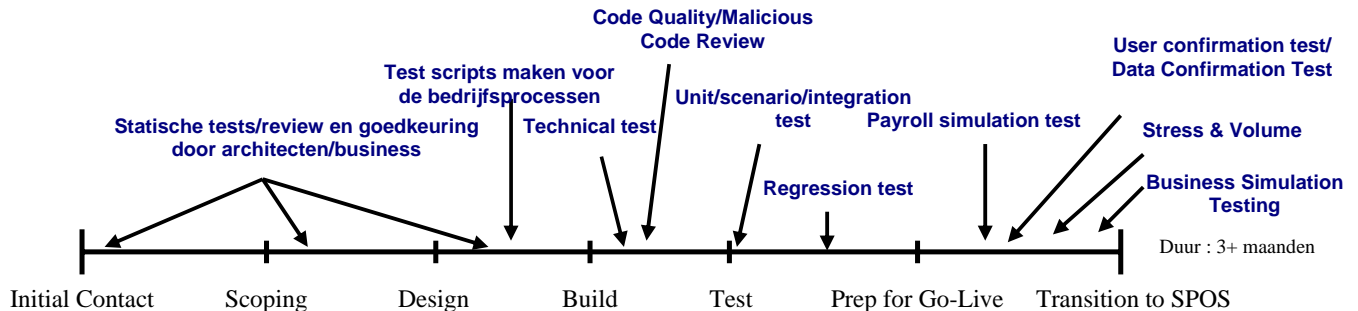
Alle verschillende testsoorten [9, w1] die worden toegepast voor de verschillende hoofdactiviteiten zijn gedefinieerd (Zie Teststrategie in bijlage C).

Hieronder zijn voor twee hoofdactiviteiten de verschillende testsoorten weergegeven die tijdens de verschillende ontwikkelingsfasen worden uitgevoerd.

Run & maintain (SPOS) – Process flow breakfix/maintenance changes



Project - Testen bij de Implementatie van een SAP systeem:



Uit de bovenstaande voorbeelden blijkt dat afhankelijk van de omvang en complexiteit van de hoofdactiviteit verschillende testactiviteiten worden uitgevoerd.

De verschillende activiteiten en soort testen zijn gedefinieerd. Per activiteit en testsoorten zijn gestructureerde interviews/vergaderingen gehouden met de volgende hoofdvragen:

Is een teststrategie geformuleerd?
Wordt voor elke test een Testplan opgesteld?
Wie voert de test uit?
Op welk systeem wordt de test uitgevoerd?
Welke soort testen worden uitgevoerd?
Welke testdata worden toegepast?
Hoe bereiken we dat het testen in de test omgeving een weergave is van de productieomgeving?
In welke omvang wordt de test uitgevoerd (testdekking)?
Hoe wordt het testen uitgevoerd?
Wat wordt gedocumenteerd (testware) en wat is de testbasis?
Welke hulpmiddelen worden er toegepast?
Hoe en door wie worden de testresultaten beoordeeld?
Wie keurt het testen goed?
Wat wordt er uiteindelijk opgeleverd na de test?
Hoe wordt de tijdige oplossing van testdefecten bewaakt?
Analyse van testdefecten.

Hoe en door wie wordt bepaald dat een systeemwijziging operationeel gemaakt kan worden?

Na het bepalen van het testprogramma per testsoort en activiteit zijn de knelpunten, die in het kader van SOx verbeterd moeten worden, direct gemarkeerd. Zie bijlage C voor het resultaat van het onderzoek van de huidige testwijze. Daarnaast is gezocht naar manieren om het software testproces nog doeltreffender te maken.

Het bepalen van de doelmatigheid en doeltreffendheid van testen is moeilijk. Hoe goed je ook getest hebt, fouten kunnen nog steeds optreden op plaatsen waar je het niet had verwacht. Testen resulterend in foutvrije software is praktisch onmogelijk.

Hoe doeltreffend wordt er nu getest? Om op deze vraag een antwoord te kunnen geven, hebben we de volgende informatie verzameld.

Hoeveel fouten worden er bij SPOS gemeld gedurende een periode nadat een nieuwe versie van het systeem operationeel is gemaakt? Vervolgens welke schade heeft dit veroorzaakt?

Uit interviews met SP medewerkers blijkt dat problemen in de productieomgeving hoofdzakelijk ontstaan omdat blijkt dat de eindgebruiker functionele gebreken heeft vastgesteld. De functionele specificaties gaven de gebruikerseisen niet goed weer, omdat eindgebruikers het nieuwe geautomatiseerde proces van te voren niet konden overzien.

Grote fouten in productie als gevolg van softwarefouten zijn in het SAP-systeem in Nederland nooit voorgekomen in tegenstelling tot de VS waar als gevolg van complexe belastingregimes onjuiste salarisberekeningen plaats vonden. Zie bijlage B voor recent voorgekomen incidenten.

Systeemfouten geconstateerd in de productieomgeving worden niet geregistreerd daarom is geen analyse mogelijk op het aantal en de aard van de voorgekomen systeemfouten.

3.2. Knelpunten

Uit het onderzoek naar de huidige testactiviteiten zijn de volgende knelpunten naar boven gekomen:

- Binnen SPP worden verschillende teststrategieën gehanteerd.
- Test scripts worden onvoldoende hergebruikt.
 - Testscripts worden niet goed beheerd.
- De diepgang van testen wordt subjectief bepaald.
- De betrouwbaarheid van gebruikte testdata laat te wensen over.
- Controle op kwaadwillende code wordt onvoldoende gericht uitgevoerd.
- Testplannen voor systeemwijzingen worden vaak niet opgesteld.
- Formele goedkeuring van het testen en de registratie daarvan vinden vaak niet plaats.
- Opslag van SOx-relevante documenten in systemen als Livelink (document management system) en Greenlight (SOx controle beheersysteem) vergt veel zoektijd voor verificatie- en afstemmingsdoeleinden.

3.3. Verbetering van de software testprocessen

Over het algemeen resulteren de verbeteringen in het kader van SOx in kleine wijzigingen in de huidige procedures. Bijvoorbeeld de goedkeuring van ontwikkelingsfasen van verzoeken voor systeemaanpassing moeten opgenomen worden als formele stap in de change management procedures.

Op basis van het vooronderzoek van de huidige testactiviteiten zijn de in paragraaf 3.2 genoemde knelpunten naar boven gekomen.

Per knelpunt is er onderzocht hoe deze knelpunten opgelost kunnen worden en ingepast kunnen worden in de huidige werkwijze en tevens de efficiëntie en effectiviteit van het testen kunnen bevorderen.

Hieronder worden de eerder gesignaleerde knelpunten verder uitgewerkt.

3.3.1. Teststrategie (Standaardisatie/harmonisatie)

Één van de bevindingen van het SOx vooronderzoek was dat er een algemene teststrategie voor alle afdelingen van SP moet komen. Een raamwerk voor het testen zal de harmonisatie van het testen voor geheel SPP ten goede komen. Bovendien kan dit leiden tot een betere communicatie tussen de verschillende afdelingen/projecten en bevordert dit de samenwerking op het gebied van testen. Een algemeen bekende standaard voor het testtraject zal ook leiden tot een betere controleerbaarheid van het software testproces. Het integreren van de teststrategie en tussenproducten voortkomend uit

gebruikte methodes/processen is essentieel voor het succes van de teststrategie. Tijdens het onderzoek naar de huidige situatie kwam al snel aan het licht dat er verwarring bestond over gebruikte testdefinities en terminologie. De nieuwe teststrategie brengt een eenduidige set van testdefinities en -terminologie.

Alle afdelingen moeten met de teststrategie vertrouwd worden gemaakt zoals genoemd in het SOx vooronderzoek.

Zie Bijlage C voor de teststrategie.

3.3.2. Hergebruik van test scripts

Test scripts worden vaak niet hergebruikt door SPOS. Implement hanteert een bibliotheek met test scripts welke niet door SPOS worden toegepast. Deze bestaat uit verschillende blokkendozen, deze blokkendozen worden met test datasets samengevoegd voor een bepaalde test. Meestal bestaat een blokkendoos uit een schermafhandeling voor een SAP-transactie.

De blokkendozen zijn gemaakt in Quicktest (geautomatiseerd testhulpmiddel) die door anderen zoals SPOS niet gebruikt worden.

Er vindt weinig of geen eigen onderzoek plaats om hulpmiddelen zelf te ontwikkelen. Er wordt uitsluitend gebruik gemaakt van pakketten, maar die zijn duur. De kosten voor het testhulpmiddel Quicktest zijn te hoog om bij SPP economisch verantwoord geautomatiseerd te testen. Dit is een obstakel voor het efficiënt testen [21].

Een geautomatiseerd hulpmiddel kan met beperkte inspanning en kosten ook zelf ontwikkeld worden.

Ik heb een demo van een geautomatiseerde testapplicatie ontwikkeld om dit te bewijzen.

Doel van het testhulpmiddel is om voor de medewerkers van SPOS makkelijker en sneller de kwaliteit van het testobject te kunnen vaststellen en direct te laten registreren dat testen zijn uitgevoerd (vereist volgens de SOx eisen).

Om dit te bereiken moet het geautomatiseerde hulpmiddel aan de volgende eisen voldoen:

- Alle interacties tussen gebruiker en SAP/Web user interface moeten geregistreerd worden.
- De applicatie moet alle geregistreerde gebruikershandelingen kunnen terugspelen.
- In de geautomatiseerde test scripts moeten controles toegevoegd kunnen worden. Bijvoorbeeld heeft de nieuwe medewerker een personeelsnummer gekregen of begint de waarde van een veld met een €symbool?
- Tijdens het afspelen van een testscript moeten alle op het scherm weergegeven waardes opgeslagen kunnen worden om later in het testscript te gebruiken. Bijvoorbeeld een medewerker wordt geregistreerd, vervolgens wordt een personeelsnummer gecreëerd dan moet dat nummer onthouden worden om daarmee wijzigingen in medewerkergegevens uit te voeren.
- Testscripts en testresultaten moeten gemakkelijk in het documentbeheersysteem Livelink opgeslagen kunnen worden.
- Per handeling moet er een schermafbeelding opgeslagen worden.
- Vanuit het geautomatiseerde testscript moet een testdocument gegenereerd kunnen worden.

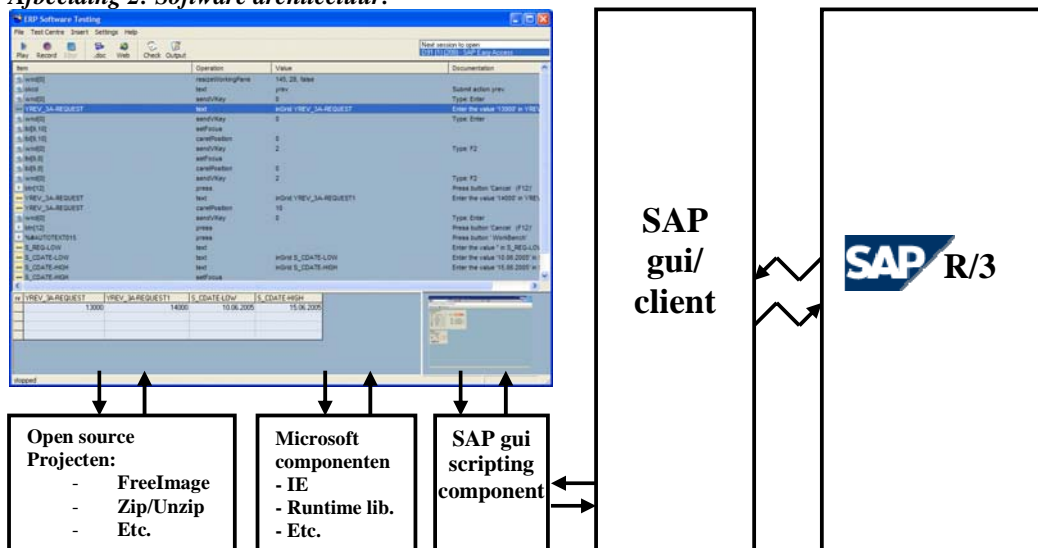
De gebruikers van het SAP-systeem gebruiken de SAP gui (graphical user interface) applicatie of een web-browser om gebruik te kunnen maken van het SAP R/3 systeem. Deze applicatie communiceert met het centrale SAP R/3 systeem waar zich de functionaliteit en database bevindt.

Met de SAP gui applicatie wordt een SAP gui scripting component meegeleverd waarmee andere applicaties met de SAP gui kunnen communiceren.

De SAP scripting component werkt volgens de COM technologie van Microsoft. Vanwege deze technologie kan het component aangeroepen worden vanuit andere applicaties geschreven in verschillende talen zoals C++, C# en Visual Basic (VB). Met C++ moet er op een laag niveau met de SAP scripting component gecommuniceerd worden, dit is complex. De COM specificaties zijn beperkt gedocumenteerd. In VB (Visual Basic) kunnen direct de methodes en attributen aangeroepen worden omdat de afhandelingen automatisch plaats vindt door de VB runtime omgeving. Omdat dit een experimentele toepassing is en in korte tijd ontwikkeld moest worden, is gekozen voor Visual Basic als hoofd programmeertaal. Daarnaast is de Visual Basic runtime omgeving reeds op alle Shell desktops/laptops aanwezig. C# of andere talen gebaseerd op .NET/CLR (Microsoft Common Language Runtime) is op de systemen goed beveiligd waardoor niet gecertificeerde .NET applicaties niet op de systemen werken, waaronder het testhulpmiddel.

Om de gewenste systeemeisen te realiseren is gebruik gemaakt van verschillende softwarecomponenten.

Abbeelding 2: Software architectuur:



De toepassing maakt gebruik van componenten die hieronder zijn toegelicht aan de hand van afbeelding 2.

Microsoft componenten	
Internet Explorer	Voor het opnemen en afspelen van test scripts voor web applicaties
Windows API	Standard Windows componenten zoals user32, GDI32 en kernel32. Deze componenten worden bijvoorbeeld gebruikt om schermafbeeldingen te maken en de gebruikersnaam op te vragen.
Microsoft Word	Gebruikt voor het genereren van het test document
Open source componenten	
FreeImage [w2]	FreeImage wordt toegepast voor het opslaan van de schermafbeeldingen in JPEG of GIF formaat en om de afbeeldingen te verkleinen zodat ze in de applicatie weergegeven kan worden. De component is ontwikkeld in C++.
Zip/Unzip Info-zip [w3]	De geautomatiseerde test scripts moeten de schermafbeeldingen, gebruikershandelingen en test documenten bevatten. Al deze bestanden worden opgeslagen in een ZIP bestand.
FTP	Component om de applicatie automatisch te vernieuwen vanuit een FTP server.
SAP componenten	
SAP gui scripting control	SAP component om gebeurtenissen te ontvangen en te communiceren met de SAP gui.

De applicatie zonder de code van de hierboven genoemde componenten bevat +/- 5000 regels code.

Een aantal medewerkers van het SPOS afdeling hebben het testhulpmiddel op verschillende wijzigingen toegepast en waren met het testhulpmiddel zeer tevreden.

Om de navolgende redenen kan het ontwikkelde testhulpmiddel de effectiviteit bevorderen voor de afdeling SPOS:

- Test scripts worden nu handmatig gemaakt maar kunnen met het testhulpmiddel geautomatiseerd aangemaakt worden. De gebruiker voert de handelingen uit en alle gebeurtenissen worden opgenomen en in het testscript geplaatst met een schermafdruck. Betere documentstandaardisatie omdat alle geautomatiseerde testscripts op dezelfde manier worden opgesteld, terwijl de handmatige testscripts zijn opgesteld door de tester.
- Eerder is besloten dat alle testscripts schermafdruckken moeten bevatten zodat te zien is dat er inderdaad getest is. Dit is een vereiste in het kader van SOx. Met het testhulpmiddel worden automatisch schermafdruckken per handeling gemaakt.
- In de standaard SPOS-procedure moet er een testdocument aan de wijziging gekoppeld worden, anders wordt de wijziging niet doorgevoerd. Het testhulpmiddel kan op basis van het geautomatiseerde test script een testdocument genereren.

- Het geautomatiseerde testscript is betrouwbaarder dan het testdocument. In het testdocument worden aannames niet altijd opgeschreven of testhandelingen in onvoldoende detail opgeschreven. Voor anderen is het dan lastig om de test volledig opnieuw uit te voeren. Met het geautomatiseerde testscript blijven alle handelingen opgeslagen.
- In de testdocumenten worden vaak niet alle testdata opgenomen terwijl het geautomatiseerde testhulpmiddel alle testdata opslaat.
- De test data kunnen gemakkelijk aangepast worden, want alle testdata bevinden zich in een tabel die bij het testscript behoort. Een testscript kan daarmee eenvoudiger hergebruikt worden omdat de testdata per testuitvoer gemakkelijk veranderd kunnen worden. Zo kunnen er onder andere makkelijker grenswaardentesten uitgevoerd worden. De testdata kunnen voor meerdere testen gebruikt worden.
- Het geautomatiseerde testscript kan gemakkelijk toegevoegd worden aan een test script bibliotheek. De gevonden test scripts kunnen beter bestudeerd c.q. geanalyseerd worden omdat deze eenvoudig afgespeeld kunnen worden. Met een handmatig testscript moet de tester de handelingen handmatig uitvoeren.
- Het testhulpmiddel zal een basis kunnen zijn voor verder automatiseren van testactiviteiten zoals bijvoorbeeld voor het genereren van test data, model-based testing etc.
- De test applicatie ondersteunt SAP gui en web applicatie interfaces. Dit zijn de twee meest toegepaste gebruikersinterfaces.

Het testhulpmiddel is geschikt voor de Customer service desk (1ste lijn ondersteuning) medewerkers, ontwikkelaars en architecten. Voor deze medewerkers zal het maken van een testdocument sneller verlopen als men de toepassing gebruikt. Per week worden er rond de 150 wijzigingen doorgevoerd die een testdocument moeten bevatten. De besparing in tijd die kan worden bereikt bedraagt op jaarbasis 0,5 man.

Het vinden van een herbruikbaar testscript moet gebruikersvriendelijk en efficiënt zijn. Daarom zal er een digitale bibliotheek moeten komen waarbij testscripts worden gerelateerd aan bedrijfsprocessen en aan SAP transacties zodat gemakkelijk de test scripts teruggevonden kunnen worden.

Training voor gebruik van de benodigde hulpmiddelen bij het opzetten en uitvoeren van testen is noodzakelijk.

Het regressie testteam zal bij opzet van een test set ook test scripts uit de bibliotheek kunnen toepassen.

3.3.3. Risicoanalyse

Testen is arbeidsintensief, daarom is er binnen SPP besloten om op basis van een gestructureerde risicoanalyse gericht te testen [10]. De uitkomst van de risicoanalyse bepaalt dan de diepgang van de uit te voeren test.

Het testen van een object zou nooit meer mogen kosten dan het financiële verlies of het verlies in 'goodwill' dat ontstaat als het object een fout bevat. Als op alle wijzigingen de volledige intensieve testprocedure (incl. goed documenteren) uitgevoerd wordt, zal dit een extra kostenpost van miljoenen per jaar betekenen. 'Risk and requirement based testing' wordt al toegepast, maar wordt nog te informeel toegepast.

De 'risk and requirement based testing' [11, 12] methode die bestaat uit de componenten 'probability of occurrence' en 'severity of the event' wordt al in grote lijnen toegepast maar moet nog formeler en geïntegreerd in de andere processen worden toegepast. De criteria voor een hoge risicowijziging moeten formeler gedocumenteerd worden. Nu wordt vooral op basis van individuele kennis en ervaring het risico bepaald.

Er zal een gestructureerde risicomatrix moeten komen die een richtlijn geeft voor de diepgang van het testen. Het toekennen van risico aan een wijziging is lastig omdat iedere betrokken medewerker (gebruiker, manager, ontwikkelaar of tester) op basis van zijn ervaring het risico anders kan inschatten. Het bepalen van een risico is subjectief [13].

Een standaard risicomatrix documenteert beter het proces zodat het risico objectiever wordt bepaald en bepaling van de omvang van de test gestructureerder tot stand komt. We hebben een algemene formule geïntroduceerd die beslissingen op het gebied van testen formeler maakt.

De formule luidt:

Risk = Impact * Chance, waarbij
Impact = loss of goodwill to use SAP * range en
Chance = probability * frequency of use.

Aan de hand van het risico en type wijziging wordt een advies gegeven welke test uitgevoerd moet worden. Deze formule zorgt voor het doeltreffender testen van wijzigingen.

Praktijkervaring van ervaren ontwikkelaars is in de waardering van de criteria verwerkt.

De risicomatrix moet een gemakkelijk te hanteren methode zijn om het risico te kunnen bepalen.

Vooraf belangrijk is, dat de risicomatrix een advies kan geven over de diepgang en omvang van het testen, zodat onnodig of te beperkt testen voorkomen kan worden.

Welk type test uitgevoerd moet worden, hangt af van het type wijziging (Zie Bijlage C).

Bijvoorbeeld voor een wijziging die meerdere transacties betreft moet een scenariotest uitgevoerd worden. De risicomatrix geeft dan aan in hoeverre getest moet worden. Een wijziging die resulteert in een interfacewijziging moet een integratietest ondergaan.

Business risk matrix

The business risk is the result of impact and chance.

Risk = Impact * Chance, waarbij

Impact = loss of goodwill to use SAP * range en

Chance = probability * frequency of use.

1.1 Impact (Loss * Range)

Loss of Goodwill to use SAP	Weight
Low: User irritated but Workaround available	4
Middle: User will/cannot complete work	8
High: Causes the user to stop using this area of SAP	15
Intolerable: Payroll/Financial/EBAS (Expatriate Basic Administrative Salary)	20

Range	Weight
Local	3
Global	5

1.2 Chance = probability * frequency

Probability	Weight
Almost impossible	0.1
Not likely: standard SAP, hardly used or no interfaces/reports	0.7
Maybe: Customized software or Interfaces and Reports	0.8
Probably: Customized software (ABAP code changed/benchmark items) and complex interfacing/reporting	1

Frequency of use	Weight
Once a year / few users	0.1
Once a month/ few users	0.4
Once a week/ many users	0.6
Once a day / many users or once a year / time-critical	0.8
Once an hour / all users or payroll run	0.9
Once a minute / all users	1

2. Type of change

Type of change
Master data
Payroll / Time schema
ABAP code - Report
Other ABAP / Workbench items
Configuration
Interface settings/partner system
Authorization/Role change

Om vervolgens te bepalen hoe er getest moet worden is per onderdeel (bijvoorbeeld: een schemawijziging met een hoog risico) gekeken wat de risicovolle plekken zijn die getest moeten worden. Welke fouten kunnen op het desbetreffende onderdeel voorkomen en welke betrouwbaarheid dient voor de wijziging nagestreefd te worden. Per onderdeel en foutsoorten is er onderzocht welke testsoorten/testtechnieken toegepast kunnen worden die dat soort fouten detecteren.

Bijvoorbeeld een hoog risico vaste gegevenswijziging kan tot gevolg hebben dat salarisberekeningen niet meer correct uitgevoerd worden of dat de transactieafhandeling niet meer correct verloopt. Daarom moet er bij dit soort wijzigingen een grenswaardentest uitgevoerd worden op de invoervelden van de transactie, gevolgd door een controle middels een salarisuitdraai op alle medewerkers waarop de wijziging invloed heeft, gebaseerd op een procescyclustest (testen van de inpasbaarheid van de procedures).

Uiteindelijk zullen deze testscripts opgenomen moeten worden in een bibliotheek waarin alle standaard testscripts opgeslagen zijn. De verwijzingen voor de testscripts worden opgenomen in de service catalogus (register met alle ondersteunde diensten die SPOS levert) waardoor per type wijziging direct de test toegepast kan worden zonder de risicomatrix toe te passen. Als het zover is zal de risicomatrix

precies moeten aangeven welke testscripts toegepast moeten worden. Dit zorgt er ook voor dat later de testactiviteiten makkelijker, gestructureerder en routinematiger verlopen en daarom eventueel in lage lonen landen uitgevoerd kunnen worden. Omdat test scripts nu nog onvoldoende hergebruikt worden zal de risicomatrix aangeven welke testscripts gemaakt moeten worden. Zie bijlage D voor de volledige risicomatrix.

Bij het verbeteren van de risicobepaling kunnen niet alleen testresultaten maar ook ervaringen en de resultaten van de Code review meegenomen worden voor het verbeteren van de risicobepaling. Dit is een iteratief proces voor verbetering van de risicoanalyse.

Voor de toepasbaarheid van de risicomatrix is het noodzakelijk dat de huidige hulpmiddelen dit ondersteunen. Er worden twee hulpmiddelen toegepast voor het verwerken van wijzigingen, namelijk ServiceCenter (change management tool) en Rev-Trac (change transport tool). Aan de tickets (meldingen en wijzigingen) die de systemen genereren zullen het bepaalde risico en de benodigde test scripts gekoppeld moeten worden voor een efficiënt toepassen van de risicomatrix. Zie bijlage G voor risicoanalyse procedure.

3.3.4. Test Data Hulpmiddel

Met de nieuwe teststrategie zal een consistenter gebruik gemaakt worden van testdata hetgeen ten goede zal komen aan de doeltreffendheid van de testen.

Functionele testen kunnen inefficiënt zijn bij het toepassen van kwalitatief slechte testdata en de schijn wekken van een kwalitatief acceptabel testobject [19].

Er bestaat daarom behoefte aan een hulpmiddel dat op efficiënte wijze een 'global test dataset' opstelt waarbij data uit productie worden geëxtraheerd en deze vervolgens zodanig aanpast worden, dat de gegevens niet terug te leiden zijn naar een medewerker of medewerkers.

Met dit hulpmiddel kunnen de testen realistischer uitgevoerd worden. En vooral in het kader van SOx zal er met betrouwbare beveiligde testdata getest moeten worden. In sommige gevallen zal het ook leiden tot tijdsparing omdat er geen testdata gemaakt hoeven te worden. Het gebruik van dit test data hulpmiddel is voor een aantal testactiviteiten verplicht gemaakt in de teststrategie.

Aanschaf van een op de markt reeds beschikbaar hulpmiddel zal binnenkort plaats vinden.

3.3.5. Software inspectie

Dit betreft een onafhankelijk onderzoek naar programmawijzigingen in het SAP R/3 systeem. De programmatuur wordt dan gecontroleerd om vast te stellen of deze programmatuur juist is geconfigureerd/gecodeerd, gebaseerd op coderingsstandaarden en naamconventies etc.

Rond 1990 werd binnen Shell IT (Information Technology) Fagan inspecties [18] geïntroduceerd.

Hoewel aanvankelijk Fagan inspecties positief werden ontvangen, werd na enige tijd geconcludeerd dat de kosten niet opwogen tegen het nut van de inspecties en werd gestopt met het houden van inspecties.

Binnen de afdeling Implement worden projecten onderzocht door SAP-werknemers, daarnaast worden controlelijsten gebruikt na ieder proces wanneer de ontwerp- en bouwfases zijn voltooid. De controlelijsten geven puntsgewijs welke controles uitgevoerd moeten worden.

Voor projecten bestaan verschillende controlelijsten die gebruikt worden om de functionele, technische en programmaspecificaties te onderzoeken op juistheid en volledigheid.

Binnen SPOS wordt programmatuur in beperkte mate door senior ontwikkelaars onderzocht op kwaadwillende code.

Er zou een meer gestructureerde methode voor uitvoering van onderzoeken (controles) ontwikkeld moeten worden.

Ontwerp- en codeonderzoeken zijn in het algemeen de meest effectieve manieren om in een vroeg stadium fouten te ontdekken [14].

In een artikel van het Royal Military College of Canada [18] wordt gewezen op een gestructureerde inspectietechniek TDI (Task Directed Inspection) waarbij de inspectie aan 8 basisvoorwaarden dient te voldoen. Deze basisvoorwaarden vinden we terug in de voorgestelde code review procedure.

Als gevolg van de snelheid waarmee systeemwijzigingen ingevoerd moeten worden, kan een volledig codeonderzoek alleen plaats vinden nadat systeemwijzigingen zijn ingevoerd.

Dit zijn activiteiten die door een onafhankelijk SPP-consultant en/of teamleider worden uitgevoerd om vast te stellen dat een gewoonlijk kritisch stuk werk van het projectteam beantwoordt aan overeengekomen specificaties en aan het gestelde doel.

Teneinde controle te kunnen uitoefenen op kwaadwillende code heb ik een controlelijst gecreëerd.

Zie bijlage E voor de controlelijst. De controlelijst bevat een reeks van vragen zoals:

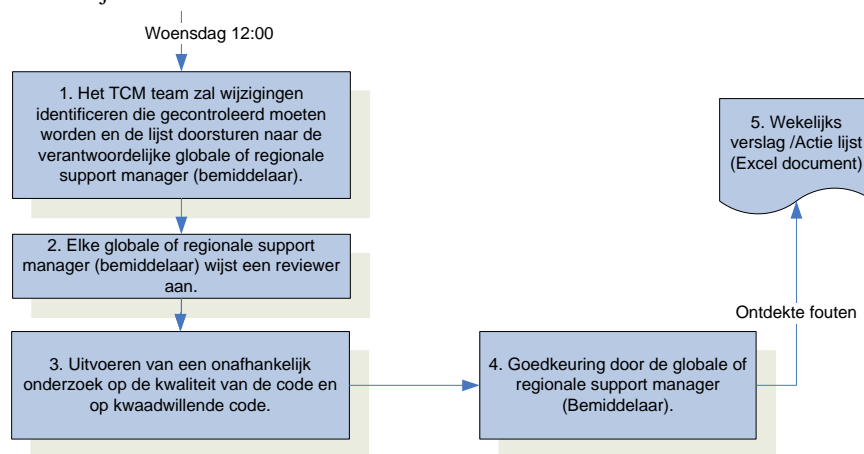
- Voldoet de transactienaam en de transactieomschrijving in het SAP-systeem aan de naamgevingstandaard?
- Vinden er geen directe database lees- en schrijfhandelingen plaats?
- Vindt er op alle invoervelden een validatie plaats?

De vragen zijn zo expliciet mogelijk opgesteld zodat het voor een ieder eenduidig is. De vragen zijn zo gedefinieerd dat bij een negatief antwoord dit direct tot actie leidt en later geautomatiseerd kunnen worden als testen middels de SAP ABAP 'code inspector' module gebeurt. Code inspector is een nieuw onderdeel van SAP R/3 4.7 dat standaard code-inspectietesten aanbiedt zoals het controleren dat er niet directe schrijf- en leeshandelingen op de database plaatsvinden. Naast de standaardtests kunnen er zelf gemaakte testen toegevoegd worden. Zodra de nieuwe release (SAP 4.7) in gebruik wordt genomen kunnen de meeste vragen toegevoegd worden aan de geautomatiseerde code-inspectie applicatie.

Het nadeel van een geautomatiseerde oplossing is dat er eventueel een uitgebreide lijst van potentiële waarschuwingen voor onjuistheden kan ontstaan. De code inspector weet namelijk niet wanneer een verdachte softwareconstructie een fout is dan wel een valide oplossing of kwaadwillende code. De lijst moet door een ontwikkelaar gecontroleerd worden en kan tot een ongewenste hoeveelheid additioneel werk leiden.

Als de onderzoeker van de wijzigingen 'nee' op een vraag antwoordt zal dat als probleem geregistreerd worden en een ander team zal verder onderzoeken hoe het probleem voortaan al dan niet verholpen kan worden.

Procedure wekelijks codeonderzoek:



Deze controlelijst zou gebruikt moeten worden door een senior peer onderzoeker om de uitkomst te rapporteren aan de globale/regionale manager. Zie bijlage F voor de software inspectie procedure die hierboven globaal is weergegeven.

De opgestelde controlelijst zal per week toegepast worden op het volgende aantal willekeurig bepaalde wijzigingen:

3 Enhancements per week, 5 emergencies per week, 2 breakfixes per week

Medewerkers die fraude willen plegen zijn moeilijk daarvan te weerhouden. Volgens onderzoek wordt de meeste fraude gepleegd door eigen medewerkers. Als er fraude wordt gepleegd zal dit uiteindelijk wel ontdekt worden. De vraag is, hoe intensief wil je controles toepassen en tegen welke prijs. De huidige procedure voor doorvoeren van wijzigingen is al complex en omvat veel controles. De controle op kwaadwillende code mag daarom niet veel tijd en geld in beslag nemen. Een controle die altijd wordt uitgevoerd is het controleren van de wijzigingen op de salarisberekening, maar per release zijn er zoveel wijzigingen dat alleen wijzigingen die een materiele invloed kunnen hebben, getest worden.

De checklist is uiteindelijk niet de meest praktische, maar is wel het best hanteerbaar omdat het andere alternatief (SAP code inspector of zelf ontwikkelde code onderzoeker) zorgt voor nog meer onderhoud van code/systemen.

3.3.6. Testplan

Vooraf voor projecten van geringe omvang en voor relevante onderhoudsaanpassingen die een subset van de roadmap (SP ontwikkelmethode) gebruiken, zal er een Test plan gedefinieerd moeten worden in het project charter. Welke testen door wie worden uitgevoerd, is zeer afhankelijk van de aard van de wijziging. Een wijziging kan bijvoorbeeld getest worden als een systeemwijzigingprocedure. Het opstellen van een testplan voor geringe projecten is nu een activiteit van de ontwikkelmethodologie (Roadmap) geworden.

3.3.7. Formele goedkeuring

Controleerbaarheid is een van de meest belangrijke aspecten van de SOx-richtlijnen. Alle goedkeuringen en paraferingen moeten traceerbaar zijn. Testen uitgevoerd naar aanleiding van een project charter of risicoanalyse moeten geregistreerd worden en goedgekeurd worden op basis van een teststrategie. De teststrategie beschrijft de procedure die uitgevoerd moet worden.

Goedkeuring van documenten werkt op dezelfde manier en zal opgevolgd worden door het TCM (Technical Change Management) team.

Alle testresultaten moeten goedgekeurd worden door een persoon die de tests niet heeft uitgevoerd. Goedkeuringen (elektronisch of handmatig) worden in verschillende IT beheersystemen (Rev-Trac, Livelink, ServiceCenter) of emails geregistreerd en opgeslagen. De locatie waar de documenten zich bevinden moet altijd toegankelijk zijn voor de eindverantwoordelijken en auditors.

4. Integratie software ontwikkelhulpmiddelen

In paragraaf 3.2 ‘Knelpunten’ is de behoefte aangegeven om de opslag van SOx-relevante documenten in systemen als Livelink (document management system) en Greenlight (SOx controle beheersysteem) te verbeteren door integratie van deze systemen.

De huidige situatie vergt veel zoektijd voor verificatie- en afstemmingsdoeleinden.

Het doel van dit onderzoek is daarom na te gaan welke mogelijkheden er bestaan om informatie uit te wisselen tussen de systemen Livelink (document management systeem) en Greenlight (SOx control systeem). Daardoor is het bijvoorbeeld mogelijk om wijzigingen in projectdocumentatie, zoals planning automatisch te verwerken in Greenlight. Dit onderzoek richt zich op het technische aspect van de communicatie tussen de twee systemen.

Livelink heeft twee mogelijkheden om te communiceren met andere systemen:

LAPI

Livelink Application Programming Interface, Dit is een bestand waarmee alle functies aangeroepen kunnen worden vanuit een zelfgemaakte applicatie, geschreven in C++ of Java of andere talen. Via het LAPI-object kan bijvoorbeeld een document opgehaald en opgeslagen worden of bestanden kunnen in folders gelezen worden etc.

Workflow

Binnen Livelink kunnen er workflows ontwikkeld worden. In de workflow kunnen een aantal stappen gedaan worden, bijvoorbeeld een formulier dat door een medewerker(s) ingevuld dient te worden of een goedkeuring dient gegeven te worden, emails of documenten die opgestuurd of opgeslagen moeten worden etc. afbeelding 3 geeft een proces van documentafhandeling door middel van een aantal stappen weer.

Een van de mogelijke stappen is de XML exchange, met deze stap kan er via XML gecommuniceerd worden met andere systemen.

Uit tests blijkt dat door verschillende toepassingen op het gebied van netwerkbeveiliging niet gecommuniceerd kan worden met een andere server. Om communicatie mogelijk te maken moet elk nieuw document via een ‘workflow’ in Livelink geplaatst worden zodat het verwerkt kan worden door een ander systeem. Dat betekent dat er per document een workflow met alle gegevens waar de documenten geplaatst moeten worden, aanwezig moet zijn.



Afbeelding 3 Livelink Workflow

Beide communicatiemogelijkheden zijn niet toepasbaar.

LAPI vraagt teveel onderhoud en kennis op het gebied van 3^{de} generatie programmeertalen.

De standaard workflow functionaliteit is te beperkt en dient aanzienlijk aangepast te worden. Wat het vooral lastig maakt is dat documenten en folderstructuren naast de workflow veranderd kunnen worden. Bij wijzigingen moet dan gekeken worden of de workflow nog wel werkt.

Greenlight is een web-applicatie van Deloitte en is geplaatst op een server van Deloitte. Greenlight is geschreven in ASP (Visual basic script voor IIS web servers). De databasestructuur is eenvoudig waardoor gemakkelijk een webservice applicatie opgezet kan worden die ontvangen data verwerkt in de database of terugstuurt. Deze applicatie is niet toegankelijk om gewijzigd te worden, omdat het van een derde partij is.

Conclusie

Vanwege beperkingen in de beveiliging en het onderhoud dat waarschijnlijk veel tijd zal gaan kosten is het niet haalbaar om de twee systemen te laten communiceren. Tevens is de workflow module van livelink te beperkt voor toepasbaarheid bij SPP en LAPI te complex. Daarnaast levert Greenlight geen standaard communicatie formaat aan zoals XML.

5. Resultaten/Conclusie

Het onderzoek heeft geresulteerd in de oplevering van een standaard globale teststrategie voor de gehele Shell People Programme organisatie, die voldoet aan de SOx eisen inzake software testing. Hierdoor ontstaat er in de gehele organisatie een betere harmonisatie op het gebied van software testing.

De verschillende modellen zoals Testability Maturity Model (TMM), Test Process Improvement (TPI) en Test Improvement Model (TIM) om het software testing proces te verbeteren of de volwassenheid vast te stellen zijn zeer beperkt toepasbaar omdat SP een unieke omgeving is die concrete oplossingen vereist. De modellen zijn te algemeen om toegepast te worden [15].

De teststrategie is ontwikkeld door gestructureerde interviews, waar per activiteit en testsoort een testprogramma is bepaald.

Het onderzoek naar de huidige werkwijze van testen voor de verschillende afdelingen gaf een duidelijk beeld van de testactiviteiten per ontwikkelproces en hoe er getest wordt over de gehele organisatie. Dit bevorderde het bepalen van hoe er voortaan getest moet worden.

De teststrategie is op 31 mei 2005 goedgekeurd door het Shell People Leadership Team en is nu officieel de teststrategie voor SPP geworden.

Als resultaat van het onderzoek naar de teststrategie zijn er een aantal knelpunten geïdentificeerd. Knelpunten zijn testonderdelen die in de praktijk niet naar wens verlopen. De nieuw gedefinieerde teststrategie lost de geïdentificeerde knelpunten op. De belangrijkste knelpunten en de gevonden oplossingen zijn hieronder beschreven:

- Risico analyse

De diepgang van het testen wordt bepaald door de kans van falen en de mogelijke gevolgen. Om dit voor medewerkers van SPOS makkelijk te kunnen bepalen is er een risicomatrix gemaakt specifiek voor SPOS. In combinatie met het Service register met alle services die SPOS aanbiedt kunnen er standaard testscripts toegevoegd worden die voor de desbetreffende wijziging van invloed zijn. Doeltreffender testen zodat risicovolle wijzigingen voldoende getest kunnen worden. Consistenter toepassen van diepgang van testen met een verminderde kans op fouten en geeft een richtlijn voor de ontwikkelaar voor het testen.

- Code onderzoek

Wekelijks worden er een aantal wijzigingen gecontroleerd door middel van een controlelijst. De controlelijst bevat vragen die moeten controleren of de wijzigingen geen kwaadwillende code bevat en kwalitatief correct zijn doorgevoerd.

De code onderzoeken moeten de kwaliteit van de code (wijzigingen) op een acceptabel niveau houden waardoor de systemen beter in de toekomst te onderhouden en te ondersteunen zijn en kwaadwillende code gedetecteerd wordt.

- Test hulpmiddel

Voor het testen van wijzigingen wordt het ontwikkelde test hulpmiddel toegepast. Het test hulpmiddel genereert testscripts en testdocumentatie. De ontwikkelde testapplicatie zal de efficiëntie van het testen duidelijk verhogen. Het hulpmiddel is gemakkelijk toe te passen.

Bij het zoeken naar de oplossingen voor het verhelpen van de knelpunten is gebleken dat de oplossingen zoveel mogelijk in de huidige processen moeten passen. Een sterke integratie is cruciaal om toepassing haalbaar te laten zijn. Invoering van de oplossingen leidt tot een doelmatiger en doeltreffender wijze van testen. Het gevaar van toepassen van verschillende methodieken/modellen is dat er 'over engineering' kan plaatsvinden. Teamleden richten zich dan te sterk op de methodiek/model, op tussenproducten (bijvoorbeeld de documenten) en te weinig op het uiteindelijke testen.

Door het beperkte aantal artikelen over ERP en testen van complexe omvangrijke bedrijfsproces ondersteunende systemen was het lastig om wetenschappelijk ondersteunde oplossingen aan te bieden. Er is eerder onderzoek [16] gedaan naar het SAP HR systeem op het gebied van testen. Deze bevindingen zijn meegenomen in de oplossingen.

In veel boeken/artikelen wordt vaak aangenomen dat de applicatie geheel zelf ontwikkeld is en de code toegankelijk is voor de ontwikkelaars (Zoals in [17]) hetgeen bij SP niet het geval is.

Het onderzoek naar de integratie van twee systemen (Livelihood en Greenlight) heeft geconcludeerd dat een samenwerking niet mogelijk is.

6. Evaluatie

Shell People Programme is een zeer internationaal gerichte organisatie. Daarom was Engels hoofdzakelijk de voertaal tijdens het uitvoeren van het onderzoek.

De introductie in de organisatie verliep zeer goed. Voor aanvang van de start van het masterproject was er al de gelegenheid om verschillende medewerkers te ontmoeten om de testproblematiek te bespreken. De eerste dag was alles reeds zodanig geregeld dat ik toegang had tot de benodigde systemen. Verschillende cursussen voor nieuwe medewerkers heb ik bijgewoond om de organisatie snel te leren kennen.

Over het algemeen worden bij SPP testen zeer professioneel uitgevoerd. Wel zijn door de SOx review een aantal knelpunten naar boven gekomen die hogere eisen stellen aan het software testproces. Oplossingen van de knelpunten zal het testproces verbeteren. De acceptatie van de oplossingen die ik heb aanbevolen heb, vereist dat deze goed passen in de huidige procedures. De betrokkenen die de oplossingen moeten toepassen gaven goede medewerking en waren tevreden met de oplossingen.

Begeleiders bij SPP spraken hun waardering uit voor de ontwikkelde teststrategie en de voorgestelde oplossingen.

SPP voldoet nu aan de SOx-eisen die zijn gesteld voor het op verantwoorde wijze testen van applicaties.

Het ontwikkelde testhulpmiddel is aan betrokkenen gedemonstreerd en vervolgens door hen getest. Opmerkingen voortkomend uit het testen van het testhulpmiddel betroffen voornamelijk:

- De response van de SAP-gui tijdens het opnemen van de test scripts was traag.
- De bestanden met test scripts en test documentatie waren omvangrijk.
- De noodzaak van training voor business analisten met weinig IT-ervaring werd als noodzakelijk gezien.

De bovengenoemde opmerkingen zijn naar tevredenheid opgelost door respectievelijk:

- Op te nemen transacties in een wachtrij te plaatsen en verwerking door het testhulpmiddel op een later moment te laten plaats vinden terwijl SAP ongehinderd verder kan gaan.
- Schermafdrucken alleen te produceren wanneer het scherm wijzigt. De schermafdrucken worden gecomprimeerd in GIF-formaat. Dit leidde tot een aanzienlijke vermindering van testdocumentatie.
- Een presentatie te ontwikkelen voor training van Business analisten.

De Business analisten toonden zich tevreden met het testhulpmiddel en ze onderschrijven de voordelen die met het testhulpmiddel bereikt kunnen worden.

Tijdens het project heb ik ervaring opgedaan in SAP HR, ABAP en testen in de (risicovolle) SP-omgeving.

Het onderzoek naar de integratie tussen twee systemen (Livelink en Greenlight) heeft tamelijk wat tijd gekost. Achteraf kan gezegd worden dat al eerder geconcludeerd had kunnen worden dat integratie niet realiseerbaar is.

7. Literatuur

Boeken/Artikelen

- [1] Michelle L. Kaarst-Brown, Shirley Kelly. "IT Governance and Sarbanes-Oxley: The Latest Sales Pitch or Real Challenges for the IT Function?", *hicss*, vol. 8, no. 8, p. 236a, Proceedings 2005, IEEE Computer Society.
<http://doi.ieeecomputersociety.org/10.1109/HICSS.2005.361>
- [2] Hans van Vliet, **Software Engineering: Principles and Practice**, 2nd edition, John Wiley & Sons, 2000, <http://www.cs.vu.nl/~hans/book.html>
- [3] Celeste See Pui Ng, Taizan Chan, Guy G. Gable, **A client-benefits oriented taxonomy of ERP maintenance**, IEEE International Conference on Software Maintenance (ICSM'01), November 2001, IEEE, pp. 528,
<http://doi.ieeecomputersociety.org/10.1109/ICSM.2001.972766>
- [4] Celeste See Pui Ng, Guy Gable, Taizan Chan, **An ERP Maintenance Model**, 36th Annual Hawaii International Conference on System Sciences (HICSS'03), January 2003, IEEE, pp. 234b,
<http://csdl.computer.org/comp/proceedings/hicss/2003/1874/08/187480234b.pdf>
- [5] Mark Keil, Amrit Tiwana. "Beyond Cost: The Drivers of COTS Application Value", *IEEE Software*, vol. 22, no. 3, pp. 64-69, May/June 2005
<http://doi.ieeecomputersociety.org/10.1109/MS.2005.58>
- [6] Bert Klok, **TErp en SAP R/3**, Sogeti Nederland B.V., Feb. 2002,
<http://www.sogeti.nl/index.html?iospagina.cfm?uNr=599>
- [7] Mark Last, Menahem Friedman, Abraham Kandel, **The Data Mining Approach to Automated Software Testing**, Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining 2003, Pages: 388 – 396, Year of Publication: 2003, ISBN:1-58113-737-0,
<http://portal.acm.org/citation.cfm?id=956795>
- [8] Tim Koomen, Rob Baarda, **TMap Test Topics**, 2004, Sogeti Nederland, ISBN: 90-72194-70-5
- [9] Martin Pol, Ruud Teunissen, Erik van Veenendaal, **Testen volgens TMap**, 2^o druk, 634 pagina's, 1999, Sogeti Nederland, ISBN 9072194713, www.tmap.net
- [10] Melinda-Carol Ballou, **Putting the "Value" Into Prioritized Testing**, Integration & Development Strategies, META Practice, 23 March 2005,
<http://www.metagroup.com/us/displayArticle.do?oid=51777>
- [11] Bob van de Burgt, Iris Pinkster, **Succesvol testmanagement: een integrale aanpak**, 2003, LogicaCMG, Ten Hagen en Stam, 422 pagina's. ISBN 9044005545
- [12] Felix Redmill, **Theory and practice of risk-based testing**, Software testing, verification and reliability, John Wiley & Sons, Volume 15, Number 1, March 2005
<http://www3.interscience.wiley.com/cgi-bin/fulltext/109792660/PDFSTART>
- [13] Felix Redmill, **Exploring risk-based testing and its implications**, Software testing, verification and reliability, John Wiley & Sons, Volume 14, Issue 1, Pages 3 - 15, 23 Jan 2004, <http://www3.interscience.wiley.com/cgi-bin/fulltext/107063125/PDFSTART>
- [14] Oliver Laitenberger, **Studying the Effects of Code Inspection and Structural Testing on Software Quality**, Fraunhofer Institute for Experimental Software Engineering, IEEE, p. 237, 1998.
<http://citeseer.ist.psu.edu/90243.html>



- [15] Macro Dekkers, **Standaardmodellen volstaan niet bij optimaliseren testproces**, Computable, 19 mei 2000, nr 20, pag 52, <http://www.computable.nl/artikels/archief0/d20ra002.htm>
- [16] **Comprehensive and effective testing of SAP HR**, Insight Consulting Partners research, http://www.insightcp.com/res_18.htm
- [17] Rick D. Craig and Stefan P. Jaskiel, **Systematic software testing**, Artech House Publishers (May 1, 2002), ISBN 1580535089
- [18] Diane Kelly, Terry Shepard, **Eight maxims for software inspectors**, Software testing, verification and reliability, John Wiley & Sons, Volume 14, Issue 4 , Pages 243 – 256, 15 Jun 2004, <http://www3.interscience.wiley.com/cgi-bin/fulltext/109060944/PDFSTART>
- [19] Elfriede Dustin, **Effective Software Testing: 50 Specific Ways to Improve Your Testing**, Dec. 2002, Addison Wesley, ISBN 0201794292
- [20] Tim Koomen, **Testing ERP implementations**, Sogeti Nederland B.V., nov. 2002, <http://www.sogeti.nl/index.html?iospagina.cfm?uNr=599>
- [21] Annukka Mäntyniemi and Pekka Mäki-Asiala, **Improving Efficiency of Testing with Test Reuse: Development of Reusable Test Assets**, In Proceedings of the First International Workshop on Quality Assurance in Reuse Contexts (QUARC 2004).

Webpagina's

- [w1] Center for Software Testing Education & Research - <http://www.testingeducation.org>
- [w2] FreeImage – <http://freeimage.sourceforge.net/>
- [w3] Info-zip - <http://infozip.sourceforge.net/>

8. Bijlagen

Bijlage A - Verklarende woordenlijst

ABAP	Advanced Business Application Programming
A&D	Architecture and Development
API	Application Programming Interface
ASP	Active Server Pages
C++	Object georiënteerde programmeertaal
C11	Register inzake controles rond ontwikkeling/onderhoud van software applicaties
CMMI	Capability Maturity Model Integration
CoBit	Control Objectives for Information and related Technology
COM	Component Object Model
CWI	Centrum voor Wiskunde en Informatica
EBAS	Expatriate Basic Administrative Salary
ERP	Enterprise Resource Planning
FTP-server	File Transport Protocol –server
GIF	Graphic Interchange Format
Global Livelink	Knowledge management system
Greenlight.	SOx web applicatie, welke documentatie omtrent registers omvat.
GUI	Graphical User Interface
IE	Internet Explorer
ISO	International Organization for Standardization
ISO 9001	ISO Kwaliteitsnormen
IT	Informatie Technologie
ITIL	Information Technology Infrastructure Library
ITS	Internet Transaction Server (SAP GUI for HTML)
Java	Object georiënteerde programmeertaal van SUN.
LAPI	Livelink Application Programming Interface
Rev-Trac	Beheer van veranderingen door het SAP landschap
SAP	Systeme Anwendungen & Produkte
SAP HR	SAP Human Resources module
SDLC	Software Development LifeCycle
ServiceCenter	Change management tool
SPICE	Software Process Improvement and Capability Determination
SPOS	Shell People Operate and Support
SP	Shell People (de Personeelsorganisatie binnen Shell)
SPP	Shell People Programme (De informatie technologie organisatie binnen SP)
SPLT	Shell People Leadership Team
SOx	Sarbanes and Oxley act
TCM	Technical Change Management
TIM	Test Improvement Model
TMM	Testability Maturity Model
TPI	Test Process Improvement
XML	Extensible Markup Language

Bijlage B – Recente grotere incidenten

Berekening en uitbetaling van een incorrect salarisbedrag aan een werknemer als gevolg van een softwarefout is in het huidige systeem niet voorgekomen. Er zijn wel incidenten gebeurd zoals bijvoorbeeld berekening van vakantiegeld/bonus op basis van een full-time baan voor parttime medewerkers. Deze fout werd ontdekt door de payroll simulatietests voordat de finale salarisverwerking plaats vond.

Een ander incident betrof een pop-up (venster) dat werd geopend door het SAP-systeem, echter omdat de medewerker via een web-based versie werkte die het venster niet weergaf, bleef de sessie altijd open staan. De server had na enige tijd geen resources meer beschikbaar om toegang te geven aan gebruikers. Dit probleem had eventueel opgelost kunnen worden door een stress en volume test. De



vraag is dan wanneer moeten we een stress- en volumetest uitvoeren, want een stress en volume test is niet gemakkelijk te organiseren en is kostbaar.

Een laatste voorbeeld, bij de implementatie van een nieuwe module voor Duitsland in de wereldwijde HR-Portal van Shell is de Portal 4 uur lang niet beschikbaar geweest. Deze Portal was uitsluitend getest in de Engelstalige versie van Internet Explorer. Bij gebruik van de nieuwe module door Duitse gebruikers bleek dat de Portal niet langer kon communiceren met de Oracle database en blokkeerde elke toegang.