

Ik ben het eens met de stelling dat business intelligence een goede exercitie is om tot waardevolle inzichten te komen. Het is echter in geen geval het 'ei van Columbus' om als organisatie compliant te raken aan allerlei geldende regelgeving.

Help, compliance ...



Bezuinigen op testproces is kwalijke zaak

Problemen moet je niet pas oplossen bij de helpdesk

Het bezuinigen op testen van software voordat deze 'in productie' wordt genomen roept vraagtekens op. De gevolgen kunnen meer schade berokkenen dat de vermeende besparing doet vermoeden. De investering in testen wordt snel terugverdiend omdat de helpdesk minder belast wordt. Daarnaast raakt de gebruiker minder snel geïrriteerd en gefrustreerd en wordt het imago van de leverancier er alleen maar beter van.

BELEID

Rob van Steenberg

De kosten van een probleem kunnen wel oplopen tot een factor honderd (Boehm 1981, Grady, 1999; NIST, 2002) als deze in de software zit en de gebruikers er mee werken. Waarschijnlijk is er een helpdesk waar de gebruikers naar toe kunnen bellen voor problemen. Vaak is er een standaard antwoord dat terug te vinden is in de beschikbare kennisdatabase. Dit is vaak voldoende om het probleem van de gebruiker op te lossen. Dit werkt efficiënt en effectief en de helpdesk medewerkers communiceren de lopende problemen met de ontwikkel organisatie. De ontwikkelaars lossen de problemen op en voeren een upgrade uit op de software bij de gebruiker en de problemen zijn opgelost.

Dan ontstaan echter weer nieuwe problemen bij de gebruiker. De

eerste keer zal de helpdesk het probleem nog moeten analyseren en rapporteren. En na een paar keer zal dit nieuwe probleem ook weer in de kennisdatabase staan, waarna de helpdesk de gebruikers kunnen helpen.

Een probleem wat in het ontwerp wordt gevonden kost zo'n honderd euro. Mocht het probleem in het ontwerp blijven - een programmeur programmeert dit in de software en de testafdeling ziet dit probleem ook niet - dan kost dit probleem duizend tot achttienhonderd euro in productie.

Waarschijnlijk komt dit verhaal bekend voor en is er een testafdeling die controleert of software goed genoeg is voor productie. Het testen van software kan opleveren dat het probleem al bekend is bij de helpdesk en dat het scenario of de workaround al klaar ligt bij de helpdesk. Dit scheelt zoekwerk van een paar mensen van de helpdesk.

Hiermee worden kostbare euro's bespaard.

Testen de ontwerpers hun eigen ontwerpen? Vast wel. De ontwerpen worden naar collega's gestuurd en deze reviewen het werk van elkaar. Misschien bespaar je hiermee wel tien problemen per ontwerp, dus (tienduizend min duizend) negenduizend euro.

Controleren programmeurs de programma code door deze te reviewen, testen programmeurs hun werk, nemen ze adviezen aan van de testafdeling? Elk probleem wordt opgelost voor tweehonderd euro per probleem, dus voor tien problemen een besparing van ongeveer (tienduizend min tweeduizend) achtduizend euro.

De testafdeling vergelijkt de ontwerpen met de geleverde software en in een productie-like omgeving en vindt hier ook weer tien problemen, die opgelost worden door de programmeurs en ontwerpen wor-

den hier en daar aangepast. Voor tien problemen bespaart je dan (tienduizend min zesduizend) vierduizend euro.

Met kwaliteitscontroles op deze niveaus wordt in dit hypothetische geval (negenduizend plus achtduizend plus vierduizend) 21.000 euro bespaard. Je bespaart daarnaast imago schade, of een mooi verhaal in de pers (Telegraaf, Computable). Dit is wat moeilijker in geld uit te drukken en na bovenstaand verhaal met de fictieve bedragen waag ik mij daar ook niet aan.

Het gaat meer om het feit dat deze kosten bij veel organisaties onder de 'verborgen kosten' vallen. Het budget van een project rekent bijvoorbeeld geen budget van de helpdesk mee. Dus alle problemen die buiten het project gehouden kunnen worden en later opgelost, vallen straks in het 'service level agreement'-budget, in het Nederlands 'service niveau overeenkomst, sno of snow

Een Agile ERP-implementatie, dat kan ook

DEVELOPMENT

Anko Tijman

'Kun je Agile ook toepassen in een erp-implementatie?' is een veel gestelde vraag aan mij nadat ik een aantal jaren geleden heb gepresenteerd.

ook bij grotere erp-implementaties toegepast worden. Veelal wordt een erp-implementatie vanuit het traditionele V-model ingestoken: in een ontwerpfase worden alle noodzakelijke aanpassingen aan het standaard systeem vastgelegd. Vervolgens wordt in een of meer

is relatief duur. Daarnaast gebeurt het ook frequent dat de gebruikers het systeem niet kunnen of willen accepteren omdat de geleverde functionaliteit niet de gewenste functionaliteit is. Bij een agile aanpak van een erp-implementatie kunnen diverse agile practices toepas-

de goede uitgangssituatie voor de iteraties te creëren kan er voorafgaand aan de iteratie al wat voorwerk verricht worden door middel van prototyping van het erp-systeem, bijvoorbeeld in een sandbox omgeving. Daarmee kan getoetst worden hoe de beoogde functio-



10.09.2010 - 13.41 UUR MARC VAN ERKELENS

Ik merk in mijn eigen kantoor dat er beduidend minder papier in omloop is dan voorheen. Heeft te maken de ict-mogelijkheden die de auteur noemt, maar zeker ook met het flexwerk-concept. Het papierloze kantoor heeft veen impuls gekregen door het kantoorloze werken.

Wat is er met het papierloze kantoor gebeurd?



Hajo

als je het uitspreekt. Sneeuw smelt, uw budget ook.

Als je naast de registratie van bevindingen en helpdesk-calls de kosten gaat noteren, dan weet je wat een probleem de organisatie kost en aan de hand daarvan kun je verbeteringen doorvoeren in de kwaliteit van deze producten. Je kunt ook gelijk beginnen met de volgende simpele verbeteringen die direct tot resultaat leiden:

■ Hou formele reviewsessies met de ontwerpers, voorgezeten door een testmanager.

■ Hou product risico analyses in een vroegtijdig stadium, zodat iedereen zich bewust wordt van mogelijke problemen en er van te voren hiermee bezig is.

■ Geef trainingen aan programmeurs hoe om te gaan met unit testen/ontwikkeltesten.

■ Zorg voor voldoende testtools voor de programmeurs.

■ Neem personeel in dienst die de ontwerpers en programmeurs kunnen begeleiden met deze controles.

Besparen op kwaliteit controles door de slechte financiële markt of

tijdsdruk? Het snel kunnen upgraden van je software bij de gebruiker? Niet zichtbaar maken wat problemen nu werkelijk kosten? Dit alles maakt het product straks duurder dan dat van de concurrentie, terwijl je zeker duizend euro per probleem bespaart als je meer nadruk legt op de kwaliteitscontroles tijdens het gehele ontwikkelproces.

Dus niet besparen op testen, maar inzichtelijk maken en investeren, zeker in deze markt waar de concurrentie groot is en de gebruiker bewust is van kwaliteit en mondig

is geworden. En de pers luistert ook mee.

Heb je ontwerp reviews, code reviews, ontwikkeltesten, systeem- en gebruikerstesten al deel gemaakt van het standaard proces? Dan kun je gaan werken aan het efficiënter maken door bijvoorbeeld te kijken of je al aan automatisch testen kan beginnen. Maar tot aan dit punt: Blijven investeren in kwaliteit, dat levert alleen maar winst op.

■ Rob van Steenbergen, directeur Chickenwings Test Consultancy

bedrijfsprocessen onder zijn hoede en is verantwoordelijk voor de realisatie ervan tot en met de produc-



■ Anko Tijman is een expert van Computable voor het topic Development en

baseerd op zijn kennis, ervaring en competenties. Op deze manier kunnen beslissingen altijd door alle disciplines beoordeeld worden en dit voorkomt dure herstelwerkzaamheden. De gebruiker kan een apart lid zijn wat vanuit de gebruikersorganisatie (deels) is vrijgesteld voor het

tomatiseerd - testen en vervolgens kan de gebruiker er direct naar kijken en het beoordelen. Dit kan dan op fijnmaziger niveau dan voorafgaand aan de iteratie en niet vanuit de zandbakomgeving, maar vanuit de ontwikkel- of testomgeving.

De interactie met de klant kan ui-

'Product Owner' separate sessies moeten beleggen waarin de belangen worden afgewogen. Hiermee moet het configuratieteam niet belast worden. De iteratieve agile werkwijze zorgt voor meer transparantie en zekerheid omtrent de waarde van de geleverde functionaliteit. Waar