

DE ARCHITECT AAN ZET

Zorg voor een productrisicoanalyse

*auteurs: Leo van der Aalst en Rik Marselis
gebaseerd op de originele publicatie in: Release*



© 2011, Sogeti Nederland B.V. te Vianen.

Niets uit deze uitgave mag verveelvoudigd en/of openbaar worden gemaakt (voor willekeurig welke doeleinden) door middel van druk, fotokopie, microfilm, geluidsband, elektronisch of op welke andere wijze dan ook zonder voorafgaande schriftelijke toestemming van Sogeti Nederland B.V.

TMap® is een geregistreerd handelsmerk van Sogeti Nederland B.V.

Testen met

Recent spraken wij een manager van een groot electronicaconcern waarvan de directie vond dat er te veel productieverstoringen waren. De manager kreeg de opdracht de softwarekwaliteit te verbeteren door het implementeren van een testaanpak. Na invoering van de testaanpak in een project bleek al snel dat bij de testuitvoering veel fouten werden gevonden wat leidde tot heel veel herstelwerk. Uiteindelijk nam het aantal productieverstoringen duidelijk af. Maar de totale kosten en doorlooptijd van het project waren 'onverwacht' hoog.

De architect aan zet

Zorg voor een productrisicoanalyse

Testen wordt, zoals in bovenstaand voorbeeld, vaak gezien als een manier om de kwaliteit te verhogen. Maar in de eerste plaats is testen een kwaliteitsmeting. En pas op de tweede plaats worden de resultaten van deze meting gebruikt om te bepalen op welke punten de kwaliteit van het informatiesysteem verbeterd kan worden.

Wij denken dat de projectmanager hier een belangrijke rol in speelt. Immers een projectmanager focust zich vaak op de hoeveelheid bestede tijd en geld en wordt meestal niet direct afgerekend op de opgeleverde kwaliteit van het informatiesysteem. Hierdoor worden testactiviteiten te vaak pas aan het eind van een project uitgevoerd, waardoor de kwaliteit van het informatiesysteem pas aan het eind van het project wordt vastgesteld. Eventuele tekortkomingen leiden dan onmiddellijk tot uitloop van het project.

In deze rubriek laten we zien dat een architect al vroegtijdig in het project de aandacht op de gewenste kwaliteit van het uiteindelijke informatiesysteem kan vestigen. Hierbij maakt de architect gebruik van een productrisicoanalyse en van diverse toetstechnieken. Ondanks de focus van de projectmanager op tijd en geld, wordt dan toch een kwalitatief beter informatiesysteem opgeleverd. Doorgaans ook nog eens sneller en tegen lagere kosten!

Focus op snelheid leidt tot lage kwaliteit en/of hoge kosten

De opdrachtgever en projectmanager hebben als belangrijke drijfveren een korte time-to-market van het informatiesysteem en de maximale kosten die daaraan besteed mogen worden. Door deze

focus is er weinig tijd en ruimte voor het weldoordacht opzetten van de architectuur en het tijdig toetsen van tussenproducten. Hierdoor wordt aan het eind van het project de initiële snelheid afgestraft door hoge kosten om fouten op te lossen en door een lange doorlooptijd om alsnog een voldoende kwaliteitsniveau te bereiken. Dat dit nog voorkomt is erg vreemd aangezien Barry Böhm ruim 30 jaar geleden al heeft gepubliceerd dat hoe vroeger in de systeemontwikkelingslifecycle fouten worden opgelost, hoe goedkoper het totale systeem zal zijn.

Verder blijkt dat informatiesystemen veel ongebruikte functies bevatten. Zo blijkt uit onderzoek uitgevoerd door The Standish Group dat slechts 42%-67% van de oorspronkelijk gedefinieerde requirements wordt gerealiseerd en Jacobs laat zien dat van die gerealiseerde requirements tot 45% in de praktijk nooit wordt gebruikt. Een onderzoek van James Martin laat nog eens zien dat 56% van de gevonden fouten hun oorsprong vinden in de requirements dus in een heel vroege fase van het project!

Dit alles is reden te meer om vroegtijdig in een project aandacht te besteden aan de gewenste kwaliteit van het uiteindelijke informatiesysteem. Onze ervaring is dat het uitvoeren van een productrisicoanalyse en het gebruik van toetstechnieken een belangrijke kwaliteitsverhogende bijdrage kan leveren.

De architect moet de productrisicoanalyse uitvoeren

De productrisicoanalyse (PRA) wordt vaak geïnitieerd door testers die het belangrijk vinden om



Leo van der Aalst en
Rik Marselis

vast te stellen waar in het informatiesysteem de grootste risico's zitten en waar dus de meeste testspanning op gericht moet worden. Maar testers zijn echter vaak nog niet in het begin van een project aangehaakt. Dus wordt de PRA relatief laat uitgevoerd. Daarom is het beter als de architect of de business analist deze PRA al eerder uitvoert. Want de gebieden met de grootste risico's moeten al in de architectuurfase en bij het opstellen van de requirements de meeste aandacht krijgen. En evenzo moet men zich afvragen of de gebieden waarbij de organisatie nauwelijks risico loopt (dus als 'het' niet werkt heeft niemand er last van) eigenlijk wel nodig zijn. Want als niemand er last van heeft als het niet werkt waarom zou je die requirement dan realiseren? Ofwel (kort door de bocht): No Risk = No Test = No Build.

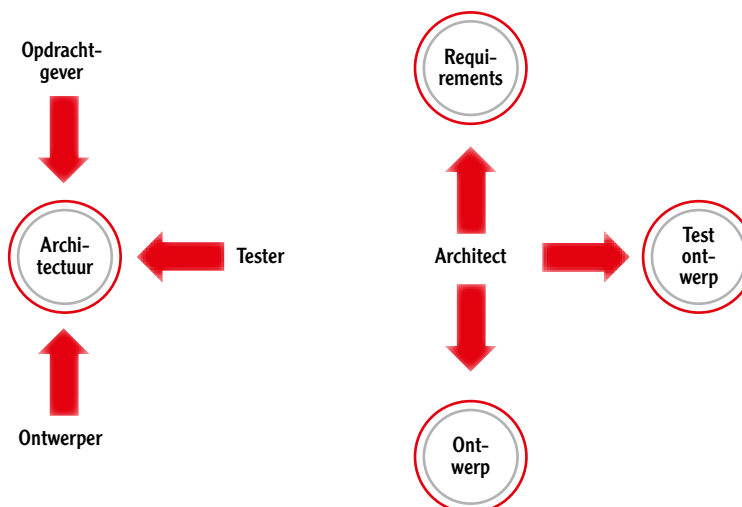
Door de vroege aandacht voor de productrisico's wordt de aandacht dus evenwichtig verdeeld op basis van het belang dat aan de requirements wordt gehecht. Bovendien kan ook de ontwikkelvolgorde zo worden ingericht dat de meest risicovolle requirements als eerste worden gerealiseerd zodat de kans dat ze op het kritieke pad komen sterk afneemt.

De architect moet toetsen en getoetst worden

Het beoordelen van tussenproducten zoals requirements-, architectuur- en ontwerpdocumenten wordt toetsen genoemd. Voor toetsen bestaan diverse technieken zoals Reviews, Inspecties en Walkthroughs. Bij alle soorten toetsen is het van groot belang om alle belanghebbenden te betrekken. Hierbij helpt het idee van "Perspective Based Reading". De verschillende belanghebbenden toetsen een tussenproduct vanuit hun eigen gezichtspunt. En juist daardoor kan heel effectief getoetst worden. Neem bijvoorbeeld een architectuurdocument (zie figuur 1). Dit moet getoetst worden vanuit de volgende invalshoeken: Opdrachtgever die de basis en randvoorwaarden heeft geschetst en wil zien of dit correct vertaald is in de architectuur. De ontwerper die het architectuurdocument gebruikt bij het maken van de systeemontwerpen. De tester die in een vroeg stadium kijkt of het nieuwe systeem wel testbaar is en of het aansluit bij de omgeving waarin het nieuwe systeem komt.

Vanuit deze zelfde gedachte is het ook belangrijk dat de architect zelf toetsingen uitvoert (zie figuur 2). De architect beoordeelt de requirements die door de opdrachtgever zijn opgesteld op haalbaarheid en consistentie. Ook toetst de architect de ontwerpdocumenten om te verifiëren of de vertaling van architectuur naar ontwerp is geslaagd en wordt het testontwerp tegen de architectuur getoetst.

Uiteraard wordt ook bij het toetsen de zwaarte van de toegepaste toetstechniek gebaseerd op het



productrisico dat verbonden is aan het betreffende tussenproduct.

Veel risico >> **uitgebreid inspecteren.**
Weinig risico >> **collegiale review.**

Kwaliteit aan het begin, snelheid aan het eind

Als de architect (bijvoorbeeld samen met de testmanager) vroeg in het project een productrisicoanalyse doet en daar de aanpak op afstemt dan wordt weliswaar in het begin van het project wat meer tijd besteed dan in de 'traditionele' aanpak. Maar doordat in het begin veel beter wordt nagedacht over wat wel en niet belangrijk is, worden de écht belangrijke requirements vroeg opgepakt en kunnen de onbelangrijke requirements buiten beschouwing blijven. Doordat de kwaliteit van de tussenproducten (te beginnen met de requirements) hoger is zal aan het eind van het project veel minder rework nodig zijn en dus wordt in die fase een hogere snelheid gehaald. Architect: pak dus deze rol op! «

Figuur 1: Architect wordt getoetst.

Figuur 2: Architect toetst.

Leo van der Aalst is research & development manager bij Sogeti en lector Software Quality and Testing aan de Fontys Hogeschool. Hij is te bereiken via leo.vander.aalst@sogeti.nl.

Rik Marselis is senior testconsultant bij Sogeti en bestuurslid van de vereniging TestNet en de stichting BNTQB. Hij is te bereiken via rik.marselis@sogeti.nl.