



Modeltechnische risico's

Bij risico's denken actuarissen (geheel terecht) vaak direct aan verzekeringstechnische risico's: hoe kunnen levensverzekeraars omgaan met de steeds stijgende levensverwachting, zijn huizen nog wel verzekeraar voor overstromingsrisico en hoe verandert het risicoprofiel van een verzekeraar nu er veel meer elektrische auto's op de weg rijden? Aan de andere kant is er ook veel aandacht voor Non-Financial Risks, zoals reputatierisico's en de impact van gebrekkige data. Een ander voorbeeld van een dergelijk risico is modeltechnisch risico: welke risico's worden gelopen bij het gebruik van het huidige modellandschap? In dit artikel onderzoeken we deze risico's in een hypothetisch scenario waarin een verzekeraar of pensioenfonds overstapt van een Excel-landschap naar een volwassen STP-model. Hier kijken we naar de risico's vóór de transitie, tijdens de transitie en daarna.

RISICO'S VOOR MODELTRANSITIE

In de uitgangssituatie wordt voor de rapportage van de verplichtingen een Excel-landschap gebruikt, waarbij meerdere gelinkte spreadsheets voor het eindresultaat zorgen. Hier is een aantal risico's te identificeren.

Zo is er vaak sprake van een Key Person Risk. Voor complexe processen worden complexe modellen gebouwd, veelal door een ervaren actuaaris met specifieke kennis van de producten. Omdat deze kennis heel specialistisch is, is deze niet wijdverspreid, en weet alleen degene die het Excel sheet (en eventueel onderliggende macro's) gebouwd heeft hoe het precies werkt. Omdat het model belangrijk is voor de complexe bedrijfsvoering, kan een situatie ontstaan waarin een model gebruikt wordt dat niet volledig wordt begrepen, waardoor als het ware met een black box wordt gewerkt.

DOOR REGRESSIETESTS EN GOED VERSIEBEHEER TE DOEN KUNNEN VEEL FOUTEN WORDEN AFGEVANGEN

Daarnaast is het geschetste proces zeer gevoelig voor fouten. Bij meerdere Excelsheets die gekoppeld zijn moeten vaak data en assumpties worden geüpdatet, formules bijgewerkt en/of aangepast worden en runs gedraaid worden. Doordat hier geen automatische tests voor zijn, kunnen er heel makkelijk fouten in de modellen sluipen. Door regressietests en goed (handmatig) versiebeheer te doen kunnen veel fouten worden afgevangen, maar idealiter worden er bij elke wijziging periodieke tests uitgevoerd. In de praktijk is dit bij het gebruik van Excel enorm lastig.

Aangezien Excel geen volwassen programmeertaal voor bulk-berekeningen is, zal de hoeveelheid berekeningen uiteindelijk een *bottleneck* worden, waardoor vaak alsnog een extern pakket moet worden afgenomen. Hoewel deze pakketten veel uit handen nemen, zorgt het ook voor een afhankelijkheid, zowel in termen van gebruikte modellen als betaalde licentiekosten. Doordat deze modellen vaak het middelpunt van een rapportageproces zijn, is het ook moeilijk van deze modellen af te stappen, waardoor men vastzit aan een rigide systeem dat niet honderd procent aansluit bij de wensen van de gebruikers.

RISICO'S TIJDENS MODELTRANSITIE

Indien gekozen wordt het modellandschap op de schop te gooien, wordt een project opgezet om (op termijn) doorlooptijd en kosten te besparen. Tijdens deze transitie liggen er echter veel sluipmoordenaars op de loer, die ervoor kunnen zorgen dat de resulterende modelstraat niet per se beter hoeft te zijn dan de bestaande.

Zo is het belangrijk een weloverwogen doelarchitectuur te kiezen: moet het model in de cloud draaien, waarom wel/niet en zo ja bij welke aanbieder? In welke taal moet het model geschreven worden en hoe moet omgegaan worden met data(opslag)? Het antwoord op deze vragen is afhankelijk van het gebruik, de schaal en complexiteit van de

modellen en heeft een grote impact op de eindsituatie. Een verkeerde keuze van de architectuur kan er namelijk voor zorgen dat het model langzamer, slechter onderhoudbaar of complexer wordt dan nodig.

Daarnaast is het belangrijk het transitieproces goed op orde te hebben en gebruik te maken van de laatste technieken. Zo moeten er schaduwmodellen gebouwd worden en moet duidelijk vastgelegd worden welke functionaliteit wanneer gebouwd wordt en waarom. Aan de andere kant moet juist worden voorkomen dat er lokaal geprogrammeerd wordt zonder versiebeheer. Anders wordt het risico gelopen dat de doorlooptijd onnodig verlengd wordt, doordat ontwikkelaars meer bezig zijn met randzaken zoals modelversies samenvoegen, discussies over prioriteringen en het verklaren van verschillen met het oude model dan het daadwerkelijk implementeren van de nieuwe modellen.

Tot slot moet de kennis binnen het bedrijf gewaarborgd worden. Omdat een dergelijk project van tijdelijke aard is, zijn voor die periode extra middelen nodig. Externe medewerkers liggen dan erg voor de hand, maar zorgen bij terugkeer naar de normale situatie voor een kennisleegloop. Indien de kennis van de nieuwe modelstructuur niet voldoende gewaarborgd is bij de interne medewerkers, kan een soortgelijke Key Person Risk situatie ontstaan als vóór de modeltransitie.

DIT IS ECHTER NIET HET MOMENT OM ACHTEROVER TE LEUNEN

RISICO'S NA MODELTRANSITIE

Na de modeltransitie is de Excel-jungle, rekening houdend met de risico's uit het vorige hoofdstuk, vervangen door een robuust, toekomstbestendig model. Dit is echter niet het moment om achterover te leunen. Ook na de modeltransitie moet het model onderhouden worden om te voorkomen dat het model onbegrijpelijk of onbruikbaar wordt.

Zo is het bijvoorbeeld belangrijk de gebruikte packages in de gekozen programmeertaal te monitoren. Zowel in analysetalen zoals R en Python als pakketten die geschikt zijn voor krachtige rekenmodellen (bijvoorbeeld C# met een html/Javascript front-end) zijn voor-geprogrammeerde functionaliteiten beschikbaar. Er kan veel tijd gewonnen worden door zogenaamde packages te gebruiken, maar hier zitten wel risico's aan vast. Zo kunnen de packages (bedoeld of onbedoeld) kwetsbaarheden bevatten waar kwaadwillenden misbruik van kunnen maken. Zo is het belangrijk altijd de meest recente versie van deze packages te gebruiken en regelmatig te evalueren of packages nog wel gebruikt worden.

Daarnaast kan het gebruik van open source packages zorgen voor onverwachte situaties. In 2016 haalde bijvoorbeeld een open source programmeur zijn code offline, waaronder een npm package genaamd leftpad. Het verwijderen van deze package, van slechts 11 regels code,

zorgde ervoor dat honderden programmeurs een foutmelding kregen bij het runnen van hun JavaScript code. Hoewel de package zelf vaak niet gebruikt werd, waren er veel packages die afhankelijk waren van deze functie. Andere packages die weer gebruik maakten van deze packages crashten hierdoor ook. Door het web van afhankelijkheden dat ontstaan was, kon het verwijderen van zo'n klein package zorgen voor een kleine softwarecrisis. Helaas kunnen afhankelijkheden niet altijd volledig worden vermeden, maar het helpt om kritisch te zijn bij het installeren van packages, vooral als de functionaliteit relatief simpel is.

MODELLEN KUNNEN ONOVERZICHTELIJKER EN LANGZAMER WORDEN

Ook is het belangrijk het model te blijven onderhouden. Hoewel het bouwen van nieuwe functionaliteiten vaak een hogere prioriteit krijgt dan modelonderhoud, is tijdig refactoren van bestaande functionaliteit enorm belangrijk. In veel gevallen passen de nieuwe methodes niet volledig binnen de huidige structuur, waardoor modellen onoverzichtelijker en uiteindelijk langzamer worden. Hoe later het refactoren gebeurt, hoe moeilijker het wordt, en hoe meer werk het kost om het model weer werkbaar te maken. Ook regulier upgraden van software (zoals bijvoorbeeld .NET Framework) zorgt ervoor dat modellen sneller worden, de meest recente beveiligingsupdates beschikbaar zijn en meer functionaliteiten gebruikt kunnen worden. Hoe eerder dit gebeurt, hoe minder werk dit kost.

CONCLUSIE

Hoewel binnen de verzekerings- en pensioenmarkt op grote schaal modelconversies van Excel naar andere talen plaatsvinden, is dit niet zonder risico's. Zowel vóór, tijdens, als na het proces moeten actuarissen waakzaam zijn, nadenken over het proces, en scherp zijn op wat het model wel en niet moet kunnen. Zo kunnen de nieuwe technologieën optimaal benut worden, zonder dat een situatie ontstaat die minder gewenst is dan die van voor de modeltransitie. ■

R.J.M. Kroon MSc AAG (links) is Senior Consultant Actuarial Technology bij Triple A – Risk Finance; R. Katoen MSc is Practice Lead Actuarial Technology bij Triple A – Risk Finance.

