

“De papieren ketentest maakt nu standaard onderdeel uit van een projectplan”



“Voordat we deze papieren ketentest hadden, liepen projecten regelmatig in de testfase van het project uit. Nadat we deze test voor de eerste keer uitvoerden bij de introductie van een nieuw verzekeringsproduct, becijferden we dat we ongeveer 3 maanden hadden gewonnen op een totale doorlooptijd van 9 maanden. Doordat we fouten al vóór de bouw hadden opgelost, werd de software conform planning op tijd opgeleverd. Bij volgende projecten heeft de papieren ketentest zich keer op keer bewezen. Hij maakt nu standaard onderdeel uit van een projectplan.”

Om een papieren ketentest goed te laten verlopen is een begeleider nodig die samen met het team de sessie zeer gedetailleerd en grondig voorbereid, de sessie onafhankelijk faciliteert en samen met alle deelnemers het eindresultaat weet binnen te halen. Je hebt iemand nodig met flexibiliteit, inlevingsvermogen, kennis van het proces én speelsheid. Iemand die durft af te wijken van het voorbereide pad als de situatie daarom vraagt. Dus bij voorkeur iemand die zelf geen onderdeel uitmaakt van het gesimuleerde proces. Daarom maken we hierbij graag gebruik van een facilitator.”

Peter Doornbos

Programmadirecteur Nationale-Nederlanden N.V.

Processimulatie vooraf scheelt een hoop geld en doorlooptijd!

door Henri Haarmans CPF

Communicatie in een IT-landschap met veel applicaties en veel eigenaren



Een grote verzekeraar beschikt over een groot en complex IT-landschap. De architectuur die hieronder ligt noemt men binnen de IT een SOA (Service Oriented Landscape). Hierin heeft elke IT-applicatie een specifieke eigenaar die zelf verantwoordelijk is voor het opstellen van eisen en wensen van deze applicatie en voor het bouwen/aanpassen van de applicatie.

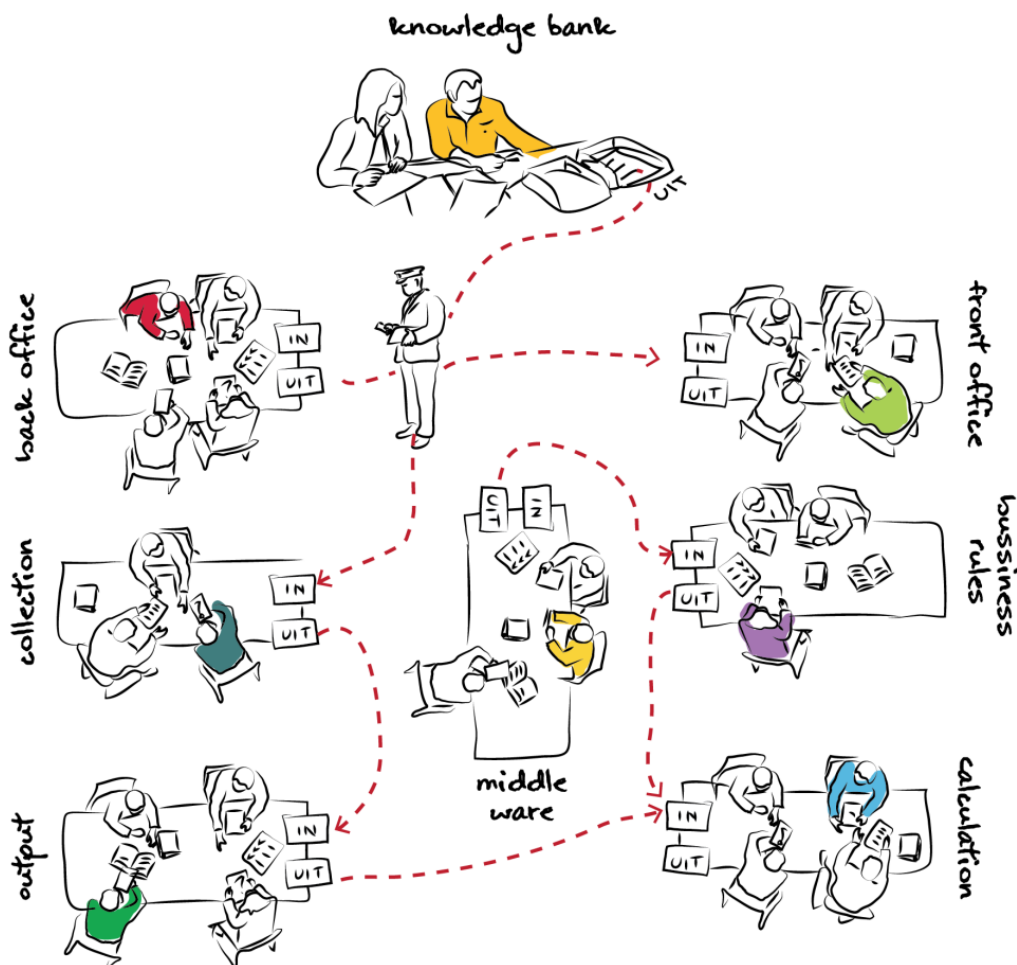
Bij deze verzekeraar gaat het om meer dan 30 eigenaren en vele tientallen applicaties. Voorbeelden van applicaties zijn: in- en excasso, web applicaties, verzekeringstechnische applicaties, output vervaardiging. Communicatie tussen applicaties verloopt via een Enterprise Service Bus (ESB), die verantwoordelijk is voor het ontvangen van berichten (services) van de ene applicatie en doorgeven aan de andere applicatie. Van dit IT-landschap is een Technical Application Architecture (TAA) beschikbaar. Hierop kan gelezen worden welke applicaties op welke wijze aan elkaar gekoppeld zijn, maar hieruit is niet af te leiden in welke volgorde en met welke informatie applicaties aangeroepen worden.

Hoe voorkom je koppelingfouten bij aanpassingen in een keten van applicaties?



Bij grote projecten worden veel applicaties aangepast. Deze aanpassingen moeten wel met elkaar in lijn zijn: de keten van applicaties moet als geheel wel (blijven) werken. Slechts een kleine kern van 2 à 3 ontwikkelaars, die overall verantwoordelijk is voor het correct laten werken van de IT-applicaties, beschikt over de complete kennis van de keten. De kans is dan ook heel groot dat de ontwikkelaars van de diverse IT-applicaties cruciale fouten maken die pas ontdekt worden bij de zogenaamde System Integration Test (SIT), waarbij de hele keten wordt getest.

Tijdens de testfase cruciale fouten oplossen is heel erg duur en kent een lange doorlooptijd. Barry Böhm heeft bewezen dat de kosten van het herstellen van fouten tijdens het IT-ontwikkelproces exponentieel stijgen met de tijd. Het ter beschikking staande Quality Management systeem van deze verzekeraar bood echter geen oplossing voor het vroegtijdig ontdekken van dit soort koppelingfouten. De vraag van de opdrachtgever was dan ook: bedenk een methode waardoor ik zo vroeg mogelijk in het ontwikkelingstraject zekerheid krijg over de correctheid van de koppelingen tussen de IT-systemen.



Simulatie van de systeem test – het ei van Columbus



Ik ontwierp een workshop met de naam Simulated System Integration Test, waarin ik de SIT zo veel mogelijk simuleerde. Dat is terug te vinden in drie zaken:

1. De groep wordt fysiek gescheiden naar applicatie. Iedere groep krijgt een eigen break-out ruimte. Op de tafel in de ruimte staan twee postbakjes: één 'in' en één 'uit'. Daarnaast liggen service-, vragen- en defectformulieren, een toelichting op de test en, indien van toepassing, een set bestandsgegevens. De centrale ontwikkelaars zijn als vraagbaak aanwezig en ook zij hebben een eigen break-out.
2. De deelnemers van de verschillende groepen mogen tijdens de testrondes niet met elkaar praten; het uitwisselen van informatie gaat via papier. De testmanager zorgt voor het transport van de 'post'.
3. Alle documentatie zoals vragenformulieren, foutdetectieformulieren, inhoud van bestanden, schermprints etc. is alleen op papier beschikbaar. Vandaar dat deze workshop ook wel de papieren ketentest wordt genoemd.

Ik regisseer het programma van de workshop strak. Het centrale team met ontwikkelaars bereid de testposten voor, meestal een stuk of 15. Deze nemen gedurende de workshop in complexiteit toe. De dag zelf heeft een vaste opzet:

- Opening: toelichting op het programma, het doel en de architectuurplaat.
- 3 testrondes: het uitvoeren van de simulatie gevolgd door een plenaire evaluatie.
- Afsluiting: het benoemen van acties en de resultaten van de dag.

In de loop van de dag zorgen energizers voor het doorbreken van de sleur.



Een testronde start met het invoeren van gegevens op een 'scherm'. Deze worden verwerkt tot services die in het uitbakje geplaatst worden. Deze post wordt opgehaald en afgeleverd bij de op het formulier met name genoemde partij. Deze gaat aan de slag, legt een serviceformulier in het uitbakje etc. Dit gaat door tot de test de gehele keten doorlopen heeft. Constateert een groep een defect, dan vullen ze een defect formulier in. Bij vragen een vragenformulier. Aan het eind van een testronde vult iedere groep een evaluatieformulier in. Deze evaluatie wordt plenaire besproken.

Doorlooptijdbesparing en een efficiënter werkproces



Het op deze manier testen van de koppelingen tussen de diverse IT-systemen zorgt er voor dat de volgende fouten in een vroeg stadium gevonden worden:

- ontbrekende of foute services;
- fouten in het datamodel tussen applicaties;
- fouten in het resterende handmatige proces;
- misinterpretaties van de eisen en wensen van de opdrachtgever;
- fouten in de inhoud van een service;
- fouten in controle functies;
- fouten in business rules.

De bespaarde doorlooptijd is afhankelijk van meerde factoren maar loopt al heel snel op van 6 weken tot 3 maanden.

Effectieve communicatie en een beter begrip van het geheel



Natuurlijk zijn de gevonden fouten een cruciaal resultaat van de workshop, want doorlooptijdbesparing is direct om te rekenen naar euro's. Daarnaast krijgen de deelnemers een gelijk beeld van het doel van het project, zien ze sneller het belang van het project en leren ze elkaar kennen. Hierdoor neemt de efficiency van de communicatie tijdens de daadwerkelijke test toe.

De business medewerkers kunnen beoordelen of het werkproces goed verloopt, ze krijgen begrip voor de complexiteit van de aanpassing en krijgen input voor het schrijven van de kennisbank en de richtlijnen. De uitgevoerde testcases kunnen 1-op-1 overgenomen worden naar de echte System Integration Test, waardoor de kwaliteit van deze test toeneemt. De testmanager leert alle betrokken medewerkers en het werkproces kennen, waardoor hij een beter begrip heeft van hetgeen hij moet testen.

Deze workshop faciliteert het tijdig opleveren van de software en een marktintroductie van een nieuw/gewijzigd product conform de wens van de opdrachtgever. Het tijdig opleveren is bij wettelijke wijzigingen cruciaal. Doorslaggevend voor het succes van deze workshop zijn twee zaken die ik als facilitator moet bewaken:

De voorbereiding door de projectgroep: de testcases moeten goed aansluiten bij de praktijk en alle mogelijke paden tussen de systemen doorlopen.

Het in stilte werken tussen de groepen onderling om de werkelijkheid zo goed mogelijk te simuleren: IT-applicaties 'praten' tenslotte ook niet met elkaar.