

## Multimedia-applicaties testen

*Ontwikkelen van de in populariteit toenemende multimedia-applicaties op CD-Rom blijft moeilijk. Door tal van verschillen met traditionele applicaties wijkt ook het testen af. Consultant P.J Koning en onderzoeker P.A. Strooper beschrijven een testmethode die speciaal werd ontwikkeld voor multimedia-applicaties. Deze is gebaseerd op de traditionele methoden en technieken.*

In januari 1996 heeft Serc (Software Engineering Research Centre) een methode ontwikkeld voor het testen van een multimedia-reisgids op CD-Rom, de Bopp Guide. De multimedia-reisgids bestaat uit een aantal databases met toeristische informatie en een engine die het gebruikersinterface verzorgt. De applicatie draait onder Windows 3.1 en Windows 95. De 'engine' is geschreven in C++ en beslaat circa 65.000 regels code. Het gebruikersinterface bestaat uit achttien soorten vensters. Op dit moment zijn er drie versies van de reisgids in gebruik: Amsterdam, Parijs en London. Dit aantal zal in de toekomst toenemen. De databases bevatten zowel toeristische als schermgegevens. Door deze opzet kan de engine generiek gebruikt worden voor de verschillende databases. Het testen van een multimedia-reisgids verschilt van het testen van traditionele applicaties. Hiervoor zijn diverse redenen. Ten eerste zijn de gegevens en de functionaliteit van de applicatie sterk gekoppeld. De gegevens vormen een integraal onderdeel van de op te leveren applicatie en moeten daarom ook getest worden. Dit in tegenstelling tot informatiesystemen waar bijvoorbeeld naw-gegevens ingevoerd kunnen worden. Hier zijn de gegevens geen onderdeel van de applicatie. Verder gaat het bij dit type applicatie om enorme hoeveelheden gegevens. De geteste multimedia-reisgids bestaat bijvoorbeeld uit drie grote databases van circa tien megabyte per stuk. Een ander verschil is dat er voor de correctheid van de gegevens nauwelijks criteria te onderkennen zijn. De juistheid van het adres van een hotel in de database is bijvoorbeeld te controleren door het te vergelijken met de feitelijke adres-gegevens. Tenslotte heeft dit type applicatie een lage graad van observeerbaarheid en controleerbaarheid waardoor het moeilijk en kostbaar is om het testen te automatiseren. Observeerbaarheid is het gemak waarmee de testuitvoer waargenomen kan worden. Controleerbaarheid is het gemak waarmee testinvoer is in te brengen. Beide zijn belangrijke test-overwegingen omdat slechte observeerbaarheid en controleerbaarheid het automatisch testen moeilijk en kostbaar maakt. Omdat dergelijke applicaties interactief zijn en een variëteit aan uitvoer leveren die moeilijk automatisch is waar te nemen, is er sprake van een lage graad van controleerbaarheid en observeerbaarheid. Vanwege de grote verschillen met het testen van traditionele systemen is een eigen methode ontwikkeld die gebaseerd is op de traditionele methoden en technieken zoals die beschreven staan in referenties [1], [2] en [4]. Referentie [3] beschrijft het ontwikkelen van multimedia-applicaties.

### Specificaties

Om een applicatie te kunnen testen moet er een specificatie van zijn waarin het gedrag en de gegevensstructuur van de applicatie beschreven worden. Deze wordt onderverdeeld in de functionele specificatie en de data dictionary die de basis vormen voor twee soorten tests. De functionele specificatie beschrijft de 'engine' aan de hand van de vensters en de mogelijke acties die in elk van deze vensters uitgevoerd kunnen worden. De 'data dictionary' beschrijft de tabellen, de attributen, de relaties met andere tabellen en de integriteitsbeperkingen ('constraints') van de database. Het testen van de 'engine' is het uitvoeren van de functionaliteitstests en de bruikbaarheidstests. De functionaliteitstest verifieert of het functionele gedrag overeenstemt met hetgeen in de functionele specificatie is beschreven [4].

De bruikbaarheidstest verifieert het programma aan de hand van gebruiksscenario's [4]. Die specificeren de manier waarop een gebruiker de door de applicatie geboden functionaliteit in de praktijk gebruikt [5]. Het testen van de databases is onder te verdelen in het testen van de integriteit en de correctheid. Vanuit het oogpunt van software-engineering is het testen van de integriteit van de gegevens het belangrijkste: de applicatie zou niet goed kunnen functioneren of zelfs volledig kunnen vastlopen bij integriteitsproblemen. De correctheid van de gegevens is van minder belang omdat correctheidsproblemen geen belangrijke consequenties met zich meebrengen.

### **Gegevens-integriteitstest**

Het testen van de grote hoeveelheid gegevens is een enorm karwei. Daarom moeten de gegevens tegen een zo gering mogelijke inspanning zo uitgebreid mogelijk getest worden. Een probleem met het testen van de database van de multimedia-reisgids is dat er voor elke stad een aparte database bestaat. Het voorkomen van herhaling is dan ook belangrijk. Om de menselijke inspanning te beperken is ervoor gekozen om de testomgeving zo op te zetten dat zo veel mogelijk tests opnieuw gebruikt kunnen worden door middel van automatisering. Het testen van de integriteit van de gegevens bestaat uit het bepalen of alle, in de data dictionary gespecificeerde beperkingen geldig zijn. Een groot voordeel van het specificeren van beperkingen is dat het testen op deze manier te automatiseren is, bijvoorbeeld door elke beperking te implementeren in een SQL-query. Zo is de constraint: 'Elke beschrijving van een "item-of-interest" moet een Nederlandse vertaling hebben' te implementeren met de SQL-query: `SELECT [BeschrijvingId] FROM BESCHRIJVING WHERE [Nederlands] IS NULL`; Deze query geeft de sleutels 'BeschrijvingId' terug voor de beschrijvingen waarvoor geen Nederlandse vertalingen zijn. De opzet van de testomgeving maakt het mogelijk om elke query apart of een aantal queries tegelijk uit te voeren en ze opnieuw uit te voeren bij een nieuwe versie van de database.

### **Functionaliteitstest**

De functionaliteitstest is het testen van de engine aan de hand van de functionele specificatie. Bij het testen van de functionaliteit van multimedia-applicaties bestaat het probleem van controleerbaarheid en observeerbaarheid. In het geval van call-based applicaties waar de toestand veranderd en geïnspecteerd kan worden door het aanroepen van functies, is de testinvoer te automatiseren met behulp van test-drivers. Bij systemen waar de invoer via het toetsenbord plaatsvindt, kan de testinvoer geautomatiseerd worden door het gebruik van een invoerbestand. In het geval van multimedia-applicaties vindt de invoer echter meestal plaats met behulp van moeilijker te automatiseren interactie, zoals een muis. Ook doen zich timing- en synchronisatieproblemen voor. De timing van bijvoorbeeld het genereren van een muis-klik in een dia-show en de synchronisatie met de laatste foto is meestal onmogelijk. Bij media zoals geluid, video en dia-shows is het moeilijk om de uitvoer automatisch te controleren. De basisfunctionaliteit van de 'engine' bestaat uit vensters waarin een aantal acties kan worden uitgevoerd. Onderdeel van de functionele specificatie is de beschrijving van de reactie op gebeurtenissen zoals het bewegen van de muis, het indrukken van een knop van de muis, de invoer via het toetsenbord enzovoort. Sommige gebeurtenissen leiden tot veranderingen in het venster - zoals de verplaatsing van de cursor -, andere tot de overgang naar een ander venster of tot veranderingen die niet zichtbaar zijn in het venster. Om het testen te vereenvoudigen wordt de functionaliteitstest onderverdeeld in passieve venster-tests die de individuele vensters testen en interactie-venster-tests die de interactie tussen de vensters testen. Tijdens de passieve venster-test worden de layout van de vensters getest en de reactie op het bewegen van de muis, zoals verandering van de cursor als deze zich in een bepaald gebied van het scherm bevindt of het tonen van help-informatie als de cursor zich op een knop bevindt. Ook

de functionaliteiten die specifiek voor één venster gelden, worden tijdens de passieve venster-test getest. Zo wordt een venster waarin hotels worden getoond, getest met nul, één, een aantal en een complete selectie van hotels. Verder wordt ook de beschikbaarheid van functionaliteit getest aan de hand van de functionele specificatie. De fax-functionaliteit is niet altijd toegankelijk. Bijvoorbeeld wel als er informatie over een hotel, maar niet als er informatie over een standbeeld getoond wordt. De interactie-venster-test probeert de interactie uit tussen de verschillende vensters en de functionaliteit van de applicatie, afhankelijk van die interactie. Zo biedt de multimedia-reisgids onder andere de mogelijkheid van het markeren van "items-of-interest" in een aantal vensters. Het bekijken van de gemarkeerde "items-of-interest" vindt in een ander venster plaats.

## **Bruikbaarheidstest**

De bruikbaarheidstest betreft het testen van de multimedia-applicatie zoals die door de consument gehanteerd wordt. Dit wordt gespecificeerd in gebruiksscenario's en vormt een onderdeel van de specificatie. Het doel van deze test is het opsporen van problemen die de consument mogelijk tegenkomt. Ook een slechte mate van gebruiksgemak kan deze test boven water krijgen, doordat bijvoorbeeld bepaalde gebruiksscenario's moeilijk zijn uit te voeren. Tijdens het testen van de multimedia-reisgids kwam ondermeer naar voren dat tijdens het ontwerp en de implementatie rekening gehouden moet worden met de testbaarheid van een applicatie. Het gebruik van een database-applicatie die SQL niet volledig ondersteunde, maakte het noodzakelijk om de databases te converteren. Door hiermee vooraf rekening te houden, kan de testinspanning voor de gegevens beperkt worden. Het bevriezen van de applicatie tijdens het testen is van belang om de oorzaak van de problemen te kunnen opsporen. Dit bleek uit het feit dat een aantal storingen dat ontstond door aanpassingen - die tegelijkertijd met het testen plaatsvonden - zeer moeilijk te verhelpen was. Vaak is het heel lastig om de implementatie tijdelijk te stoppen, doordat de druk om de applicatie op tijd op te leveren vaak erg groot is. Hiermee dient men dan ook tijdens het plannen van de ontwikkeling van een multimedia-applicatie rekening te houden. Door ook tijd voor het debuggen in te plannen, kan voorkomen worden dat een test volstrekt onzinnig wordt, omdat er geen tijd meer is om de geconstateerde problemen op te lossen. Een belangrijke keuze die bij het testen van de multi-media-reisgids is gemaakt betreft het gebruik van de database met de feitelijke gegevens in plaats van een representatieve testdatabase. Hierdoor is het tijdens de functionaliteits- en de bruikbaarheidstest lastig te achterhalen of de gevonden problemen zich in de engine of in de database van de applicatie voordoen. Tevens wordt een aantal problemen meerdere malen gesignaleerd doordat de uitgifte van nieuwe versies van de engine en de database niet synchroon lopen. Door het gebruik van een representatieve testdatabase zou de functionaliteits- en bruikbaarheidstest minder inspanning vergen en is een asynchrone ontwikkeling van de 'engine' en de database mogelijk. De extra inspanning die geleverd moet worden om een representatieve testdatabase te bouwen, weegt in dit geval niet alleen op tegen de beperking van de inspanningen tijdens de functionaliteits- en de bruikbaarheidstest maar ook tijdens het debuggen. Voor het automatiseren van de functionaliteits- en bruikbaarheidstest zijn tools op de markt waarvan er een aantal bekeken is en er één is uitgetest in het kader van de multimedia-reisgids. Het betrof een zogenaamde record-and-playback testtool dat het opnemen en afspelen van testscenario's mogelijk maakt. Uiteindelijk is het tool niet gebruikt tijdens het testen omdat de eerder geschetste timing- en synchronisatieproblemen zich voordeden bij de implementatie van de testscenario's. Hierdoor moest het testen van elke nieuwe versie van de multi-media-reisgids met de hand plaatsvinden.

## **Auteurs**

- Drs. P.J. Koning is werkzaam als consultant bij Serc in Utrecht.
- Dr. P.A. Strooper is docent van de vakgroep Computer Science aan de Universiteit van Queensland in Brisbane, Australië. Van januari tot en met juni 1996 was hij als onderzoeker werkzaam bij Serc in Utrecht.

## **Literatuur**

- [1] B. Beizer, Software System Testing and Quality Assurance, Van Nostrand Reinhold, 1984.
- [2] B. Beizer, Software Testing Techniques. Van Nostrand Reinhold, second edition, 1990.
- [3] M. Muhlouser, Issues in Multimedia Software Development, Proceedings of the International Workshop on Multimedia Software Development, IEEE Press, 1996, pag. 2-7.
- [4] G. Myers, The Art of Software Testing John Wiley & Sons, 1979.
- [5] I. Jacobson, Object-Oriented Software, Engineering - A Use Case Driven Approach, Addison-Wesley Publishing Company, 1992.