

Focus op kwaliteit met Visual Studio 2008

TIJDBESPARING UNIT-TESTEN DOOR BETERE PERFORMANCE TESTENGINE

Voorafgaand aan Visual Studio 2005 was het voornaamste thema van een nieuwe Visual Studio-release de ondersteuning van de recentste technologieën en de verbetering van de productiviteit van de ontwikkelaar. Met Visual Studio 2005 Team System heeft Microsoft een geïntegreerde omgeving geïntroduceerd die gericht is op het hele team. Het verbeteren van de samenwerking van het team, het verschaffen van meer inzicht in zowel de status van de voortgang van het project als de kwaliteit van het ontwikkelde systeem, moeten het team in staat stellen de vele uitdagingen aan te gaan op hun weg naar de eindstreep. In dit artikel bespreken we de verbeteringen en uitbreidingen in Visual Studio 2008 op het vlak van unit- en webtesten.

Het Test Framework in Visual Studio

In Visual Studio 2005 is het Test Framework geïntroduceerd. De bouwstenen van dit framework zijn gericht op de ondersteuning van de test-workflow, zoals de ontwikkeling van testcases, het managen van deze ontwikkelde testcases, een executie-engine voor het uitvoeren van de testcases en als laatste de analyse van de testresultaten. Het framework is de basis voor de uitvoering van uiteenlopende testtypes zoals de (Database) Unit Test, Web Test, Load Test en Manual Test. Door een Generic-test kunnen ook andere externe testtypes in de omgeving worden geïntegreerd; zie afbeelding 1.

Optioneel kan Team Foundation Server (TFS) inpluggen op dit framework en zo ontstaat de mogelijkheid de testresultaten te publiceren in de TFS-repository. Bovendien kun je vanuit de test-results-view een workitem vervaardigen, waarbij de testresultaten aan dit workitem worden gekoppeld. Ook is er de gelegenheid om de ontwikkelde testcases in een team-build te gebruiken en een checkin-policy te hanteren die verifieert of de ontwikkelaar zijn unit-tests heeft doorlopen.

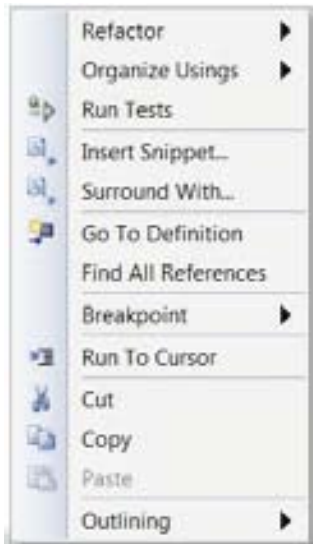


Afbeelding 1. Overzicht testtypes.

Unit-testing en Test Driven Development

De Agile-community heeft als geen ander het gebruik van unit-testen onderstreept, menig ontwikkelaar heeft unit-testen dan ook opgenomen als onderdeel van zijn gereedschapskist. De wijze waarop unit-testing echter wordt ingezet, verschilt enorm van project tot project. Er is de klassieker benadering, waarbij bestaande code als uitgangspunt wordt genomen voor de generatie van een unit-test. De intentie is om met de unit-test de juiste werking van de functionaliteit te bevestigen. Aanhangers van Test Driven Development (TDD) volgen een andere benadering. Zij hanteren het red-, green-, refactor-motto. De intentie is juist eerst vast te leggen wat jouw code behoort te doen via de unit-test, en dit vervolgens te implementeren. Men start dus met het ontwikkelen van de geautomatiseerde (unit)test die faalt, aangezien er nog geen functionele code is om te testen. De rode status is bereikt. In de volgende stap wordt deze test in een groene status omgezet, door net genoeg code toe te voegen. Vervolgens richt men zich op een ander scenario. Op identieke wijze doorloopt men het proces, daar waar nodig volgt een refactor-stap. In kleine stappen complementeert de ontwikkelaar dus zowel de code als de bijbehorende unit-test. De unit-test vervult dus meer de rol van specificatie dan een van verificatie. Hoewel de naam anders doet vermoeden, is TDD een best-practice gericht op het ontwikkelen van code, en niet zozeer op het testen. Test Driven Development is ondertussen behoorlijk populair geworden.

Visual Studio ondersteunt beide scenario's, dus zowel aanhangers van TDD als ook de klassieke benadering zullen in Visual Studio een weg vinden. Voor diegenen die de voorkeur hebben hun testcases te genereren op basis van hun code, is het goed om te melden dat de generatie ook scenario's met generics ondersteunt. Ook is er nu de mogelijkheid een testcase te genereren op basis van een assembly. Noemenswaardig zijn ook de verbeteringen ten aanzien van het gebruik van overerving bij het ontwikkelen van een unit-test. Een testclass kan dus nu members erven van andere testclasses. Dit betekent dat je nu duplicate testcode naar een base-testclass kan verplaatsen, waardoor de onderhoudbaarheid van jouw testcode verbetert. Bij gebruik van testclass-inheritance zal de Test View en Test List Editor de test-methods



Afbeelding 2. Right-click test-execution

tonen uit zowel de base-testclass als de child-classes. Je kunt de kolommen van de views zo aanpassen dat een kolom de volledige class-naam toont.

Right-click test-execution

Visual Studio 2008 heeft een aantal verbeteringen om bovenstaande scenario's beter te kunnen ondersteunen. Om een red green refactor-iteratie te doorlopen, is een snelle evaluatie van de unit-test een must, te beginnen met een snelle selectie van de uit te voeren test. Er is een aantal nieuwe mogelijkheden om te bepalen welke tests uitgevoerd gaan worden, dit mechanisme wordt right-click test-execution genoemd; zie afbeelding 2.

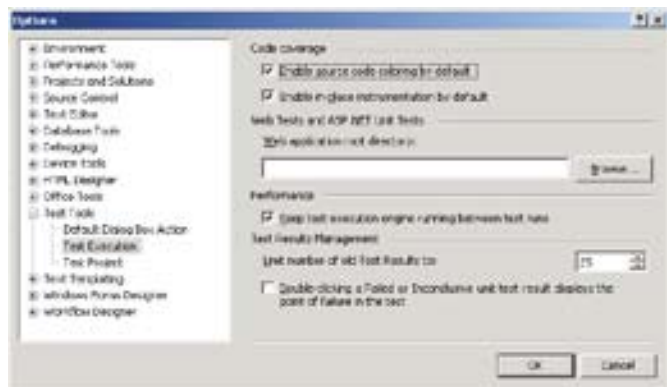
Dit kan plaatsvinden vanuit de context van de testmethode waaraan men werkt in de editor, de actieve testclass of het testproject. Afhankelijk van de gekozen context wordt dus een enkele test van een testclass, alle testmethoden in de class of alle tests in het testproject uitgevoerd. Stap twee is de daadwerkelijke executietijd van een unit-test, deze is in Visual Studio 2008 behoorlijk verbeterd.

Test results management

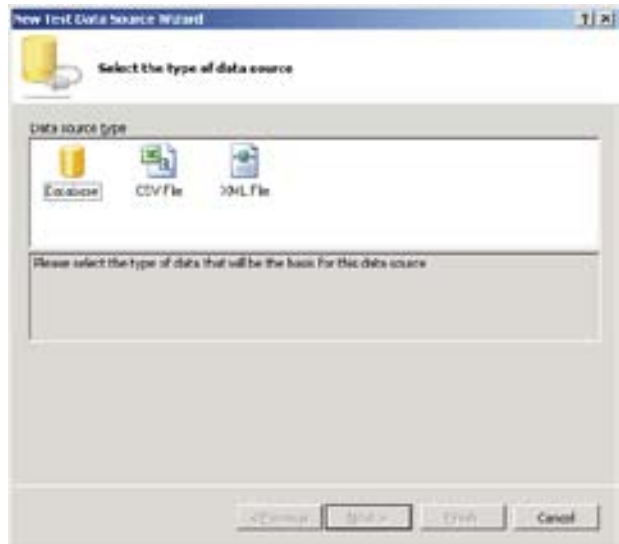
Een kleine, maar zeer aangename verbetering is de optie om het aantal oude files met de testresultaten te beperken. Het handmatig verwijderen van files met oude testresultaten behoort hiermee dus tot de verleden tijd; zie afbeelding 3.

Data-driven unit-testing

In Visual studio 2005 bestond al de mogelijkheid data-driven unit-testing toe te passen. Een specifieke test kan hiermee herhaaldelijk uitgevoerd worden met de testdata uit de gekoppelde datasource als input. Om comma-separated values (CSV) te gebruiken in Visual Studio 2005, moest de developer over de



Afbeelding 3. Test results management



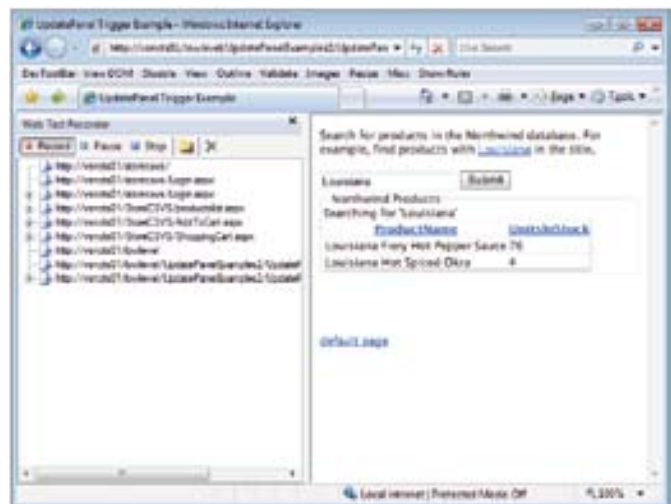
Afbeelding 4. CSV- en XML-files zijn als bron van de testdata te gebruiken

nodige kennis beschikken over hoe OLEDB over JET omging met dergelijke files. Het gebruik van XML was nog lastiger, aangezien hiervoor een custom datasource nodig was. Visual Studio 2008 maakt het nu veel gemakkelijker om CSV- en XML-files als bron van de testdata te gebruiken; zie afbeelding 4.

Een optie is om de CSV- of XM-file in jouw project op te nemen, zodat deze automatisch wordt gedeployed. Ieder lid van het team dat later de test gebruikt, kan deze zonder problemen direct toepassen. Wat valt er nog meer op te merken over unit-testing in Visual Studio 2008? Microsoft heeft goed geluisterd naar de feedback uit de community en heeft besloten de unit-testingfunctionaliteit en ook de testlist onderdeel te maken van de Pro-editie van Visual Studio.

Ondersteuning voor AJAX in de Web Test

Sinds de introductie van Web Testing in Visual Studio 2005 is er het nodige gebeurd op het vlak van innovatie rondom webdevelopment. Om webapplicaties te kunnen testen, die gebruikmaken van AJAX-technologie ter verbetering van de user-experience, moest tot voor kort additionele tooling worden ingezet, zoals Fiddler. De Web Test-recorder van Visual Studio 2008 heeft dit gat echter gedicht. Het is nu ook mogelijk AJAX-requests op te nemen; zie afbeelding 5. Fiddler kan overigens nog steeds nuttig zijn voor de diagnose van problemen bij het opnemen en afspelen van jouw webtests. In Visual Studio 2008 kun je nu door middel van selectie van een aantal request in een bestaande webtest een nieuwe webtest te vervaardigen. Je kunt ook een aanroep naar een andere webtest invoegen. Het hergebruik van een andere webtest in jouw



Afbeelding 5. Ondersteuning voor AJAX in de Webtest recorder

webtest heeft beter onderhoudbare testcode als eindresultaat. De webtest-engine maakt gebruik van zogenaamde extraction rules om data uit een response te halen. Deze data worden dan als 'name value pairs' opgeslagen in een webcontextparameter. Extraction rules kunnen omgaan met form fields, text, attributes, headers, regular expressions, en hidden fields. Een webcontextparameter kan weer gebruikt worden in een webrequest. Validation rules verifiëren dat een webapplicatie juist functioneert. Validation rules onderzoeken de pagina op de aanwezigheid van tekst, tags, attributen, form fields en hun waarden, en kunnen ook de tijd die nodig is voor de afhandeling van het request beoordelen. Visual Studio 2008 heeft nu ook ondersteuning voor testlevel validation rules. Een validation rule op testlevel werkt voor alle individuele requests in de test en wordt dus bij elke request in de test geëvalueerd.

In de 2005-release vereist het de nodige creativiteit om handelingen in te voegen, die eenmalig aan het begin en/of einde van de test plaatsvinden. Voorbeelden zijn zaken als login, checkout en het toevoegen van een account. Visual Studio 2008 heeft nu ondersteuning voor de start en het einde van een test. In de test-mix-editor kan door middel van de activering van de gewenste optie een initialisatie- en/of termination-test worden geselecteerd. Bij het uitvoeren van een we test zal een fout bij een http-request er voor zorgen dat de test faalt. Ondanks dit resultaat, zal de test niet automatisch stoppen. Door middel van het property 'Stop On Error' is het nu mogelijk om bij het optreden van een dergelijke situatie de test te stoppen. Bovendien zijn er enkele UI-verbeteringen in de Web Test-playback. The UI in playback-mode is meer afgestemd op die van de webtest-editor. Men heeft de visualisatie van het verloop van jouw testcase verbeterd. Als jouw test bijvoorbeeld een transactie bevat met daarin een aantal requests, dan zal die door een tree-structuur worden gevisualiseerd. Ook een redirected request zal als een child-node van het oorspronkelijke worden getoond. Dit zorgt er voor dat een dergelijk scenario beter is te overzien. Ook zijn er additionele buttons op de toolbar om door de requests te navigeren. Heel nuttig zijn de 'Find next error'- en 'Find Previous Error'-buttons.

Ook heel handig is de presentatie van de request-parameters. Van een request worden de headers, cookies, query string-parameters en Form Post-parameters apart gepresenteerd in tabelvorm. De oude variant waarbij alleen de ruwe data werden getoond, bestaat nog steeds, maar dit maakt het veel gemakkelijker om de inhoud van een request te analyseren. Ook nieuw is de optie om de response in een html-editor te tonen, iets wat ook bijdraagt aan een betere analyse van de response. De extensibility van de webtest-engine is in deze release ook verbeterd; ondersteuning voor het gebruik van verscheidene webtest-plugins, de overdracht van parameters en ook request level-plugins behoren tot de mogelijkheden.

Performance

In Visual Studio 2008 zijn vele features toegevoegd die er voor zorgen dat de onderhoudbaarheid van de testcode verbetert. Ook is de performance van de testengine verbeterd, waardoor je vooral veel tijd zult besparen bij het gebruik van unit-testing.

Gerard van der Pol is Developer Technical Specialist in de Developer & Platform Group van Microsoft Nederland. Hij is te bereiken via gevdpol@microsoft.com

Referenties

Guidelines for Test-Driven Development: [http://msdn2.microsoft.com/en-us/library/aa730844\(VS.80\).aspx](http://msdn2.microsoft.com/en-us/library/aa730844(VS.80).aspx)

Unit-testing walkthrough: [http://msdn2.microsoft.com/en-us/library/ms379625\(VS.80\).aspx](http://msdn2.microsoft.com/en-us/library/ms379625(VS.80).aspx)

Team Test video's: <http://www.asp.net/learn/vsts-videos/>

Testen met Visual Studio 2005: .NET Magazine #11