

Studienarbeit

Datum: 15. Juli 2010

Themenbereich: Software-Engineering / Testen von Anwendungen

Ausgangssituation:

Komponententests (engl. *unit tests*) helfen Entwicklern, die Funktionalität von Methoden zu überprüfen und lassen sich darüberhinaus als Regressionstests einsetzen. Softwareprojekte besitzen neben funktionalen häufig auch nichtfunktionale Anforderungen, die beispielsweise die Anwendungssicherheit oder die Leistung der Software spezifizieren. Insbesondere die Leistungsmessung ist mit *unit tests* nicht oder nur unzureichend möglich: Werte wie der dynamische Speicherverbrauch einer Klasse zur Laufzeit, der Cache-Datenverkehr oder die Anzahl der Synchronisierungsoperationen innerhalb einer Methode lassen sich in *unit tests* bislang nicht spezifizieren und testen.

Ziel:

Das Ziel dieser Studienarbeit ist es, das Rahmenwerk für *unit tests* von Visual Studio so zu erweitern, dass Vorgaben für leistungskritische Methoden automatisch bei jeder Testausführung überprüft werden. Falls die spezifizierten Leistungsvorgaben nicht eingehalten werden, schlägt der Test fehl.

Leistungsindikatoren, die in Tests integriert werden sollen, sind zum Beispiel:

- Windows-Leistungsdaten (vgl. Performance Monitor)
- CPU Hardware Performance Counter

Die so ermittelten Indikatoren sollen von Testern durch Assert-Anweisungen genutzt werden können:

```
Assert.LessThan(Performance.MemoryAllocated, 50, Unit.MegaByte);  
Assert.LessThan(Performance.CacheLinesEvicted, 300000);
```

Ein Test mit diesen Zusicherungen soll scheitern, wenn während der Ausführung mehr als 50 Megabyte an Speicher allokiert und gleichzeitig mehr als 300.000 Cachezeilen entfernt wurden.

Voraussetzungen:

- Kenntnisse in C#, C/C++
- Interesse an der Windows32/64-API
- Erfahrung im Testen von Anwendungen hilfreich

Betreuer:

Jochen Schimmel und Korbinian Molitorisz sind wissenschaftliche Mitarbeiter am Lehrstuhl Prof. Walter F. Tichy. Sie befassen sich im Rahmen ihrer Forschung mit den Themen „Parallelisierung sequentieller Softwaresysteme“, „Werkzeugunterstützung zur Identifizierung parallelisierbarer Regionen“, „Werkzeugunterstützung zur automatischen Testfallerzeugung“ und „Erzeugung von Testfällen für parallele Anwendungen“.

Diese Studienarbeit wird von beiden Mitarbeitern betreut.