

Anfänge der Testautomation

Harry M. Sneed
Technische Universität Dresden
SoRing Kft. H-1221 Budapest
ZTP Prentner IT, Wien
Harry.Sneed@T-Online.de

Abstrakt: In diesem Beitrag schildert der Autor die Entwicklung der ersten Testwerkzeuge im deutschsprachigen Raum. Es begann mit dem Software-Prüfstand das ab 1977 bei Siemens in München für den Test von SPL Moduln unter BS2000 konzipiert wurde. Dieses Test-Tool benutzte die Compilerlisten um die Adressen der Daten zu gewinnen und eine interaktive Dialogsprache um die Daten zur Laufzeit zu manipulieren. Das Tool wurde für den Modultest in dem integrierten Transportsteuerungssystem der Bundesbahn eingesetzt. Es folgten weitere Modultestwerkzeuge der gleichen Bauart für IBM-Assembler, IBM-8100 COBOL, PLI und Kienzle's strukturierte Programmiersprache für Mikroprozessoren. Im Jahre 1982 wurde das erste Integrationstestsystem für die Tornado Lagerhaltungssoftware auf dem IBM-DOS-VSE entwickelt. Dieses System enthielt schon alle Komponenten eines modernen Testsystems einschließlich Testdatengenerator, Testmonitor, Testergebnisvalidator, Dateikomparator und Testskriptsprache, nur alles sehr rudimentär implementiert. Die Frühphase der Testautomation endete mit der ACM Testkonferenz im März 1982 an der Bundeswehrhochschule in Neubiberg

Schlüsselwörter: Testautomation, Modultester, Testüberdeckungsmessung, Datengeneratoren, Ergebnis-Validator, Testmonitoren, Embedded Softwaretest, Integrationstestsystem, Testorakel, Testskriptsprache.

1 RXVP als Vorbild für die deutschen Testwerkzeuge

Man sage, Krieg sei der Vater aller Dinge. Das trifft im Falle der Testautomation voll zu. Es war das U.S. Verteidigungsministerium das den Auftrag für das erste Software Testsystem vergab. Dieses System mit dem ungewöhnlichen Titel „Research Test and Validation Package - RXVP“ wurde von der General Research Corporation in Santa Barbara California entwickelt. Später wurde das Projekt in SQLAB umgetauft. RXVP enthielt bereits alle Komponente eines modernen Testsystems – statischer Analysator, dynamischer Analysator, Testüberdeckungsmonitor, symbolischer Datengenerator und Ergebnisvalidator. Die Ergebnisse wurden gegen Post-Assertions geprüft. Die Eingaben wurden aus den Pre-Assertions abgeleitet. Der Testingenieur steuerte den Testablauf über ein Texas Instruments graphische Oberfläche mit Lichtstift. Er konnte den Ablauf an jedem beliebigen Netzknoten

unterbrechen und den Stand der Zwischenergebnisse überprüfen. RXVP wurde benutzt um die Kommunikationssoftware in dem Ballistic Missile Defense System zu testen. Es hat vorher einzelne Test Tools gegeben um die Daten zu generieren, den Code zu instrumentieren und die Testüberdeckung zu messen, aber RXVP war das erste Paket mehrerer integrierter Werkzeuge. Das Problem der Testwerkzeugintegration ist bis heute geblieben.

2 Der Siemens Prüfstand

Der Leiter des RXVP Projektes war ein gewisser Dr. Ed Miller der seine Doktorarbeit über Programmtesten geschrieben hat. Er studierte an der Universität Maryland wo auch der Autor her kommt. Der Autor wusste von seiner Arbeit und besuchte ihn 1977 auf dem NCCC Kongress in Dallas. Dort haben sie über eine Zusammenarbeit gesprochen. Der Autor hatte gerade als Testmanager für das Siemens I.T.S. Projekt begonnen und wollte den Test dort mit den Methoden des RXVP automatisieren. Eine Übertragung des Tools kam nicht in Frage. Es blieb nur übrig das Tool selber nach zu bauen. Der Autor hatte schon einige Bestandteile als Tester von der UDS Datenbank Software angefangen. Allerdings war er ganz allein, da es bei Siemens sonst keine Tester gab. Im Winter 1977/78 begann er die ersten SPL Module zu testen.

Auf einer Testkonferenz in London, die von Dr. Miller organisiert wurde, traf der Autor auf einen Ungar der ihm Testkapazität in Budapest anbot. Der Autor nahm das Angebot an und baute mit Hilfe des Dr. Millers ein Software-Testlabor in Budapest auf. Im Mittelpunkt dieses Testlabors stand das Tool Prüfstand. Mit Prüfstand konnte der Tester Module aufrufen und mit Daten versorgen. Den Testablauf verfolgte er über den BS2000 Bildschirmmonitor. Er konnte den test jederzeit unterbrechen und Datenzustände kontrollieren und manipulieren. Vorher wurden die Testobjekte instrumentiert so dass auch die Testpfade wie bei RXVP verfolgt werden konnten. Das Ziel war alle Pfade zu durchlaufen und mindestens 85% Zweigüberdeckung zu erreichen. Dabei sollten möglichst viele Fehler im Code aufgedeckt werden. Der Unterschied zu RXVP war jedoch das in RXVP gegen die Rnet Spezifikation getestet war. Mit Prüfstand wurde der Code gegen sich selbst getestet. Auf dieser Weise war es nur möglich bestimmte Fehlerarten zu finden. Ob der Code wirklich das tat was von ihm erwartet wurde konnte keiner

sagen. Es fehlte das Testorakel. Somit konnten nur die Hälfte der gesamten Fehler durch den Modultest gefunden werden. Es müssten andere Tests folgen.

Mit einem ähnlichen Ansatz wurde das Quelle Software Testsystem – QSTS – für den Test von IBM-Assembler Module entwickelt. Dieses Tool lief jedoch im Batchbetrieb. Die Tester mussten im Voraus die Testfälle mit einer Skriptsprache spezifizieren. Dazu gehörte die Voraussage der Testergebnisse die gegen die Nachbedingungen geprüft wurden. Der Benutzer – die Firma Quelle – setzte eine semi-formale Spezifikationssprache ein um die Modulein- und ausgaben festzulegen. Das hat dazu geführt dass auch Abweichungen vom Sollverhalten notiert wurden. Mit diesem Tool wurden die Testskripte in Assemblermodule umgesetzt und als Testtreiber und Teststubs mit dem Testobjekt zusammen gelinkt. Dadurch wurde der Modultest als ein einziger Job im IBM-OS Betrieb ausgeführt. Der Tester bekam die Ergebnisse erst am Ende des Tests zu sehen. Bei Prüfstand konnte er schon während des Tests den Ablauf verfolgen und auch eingreifen. Der IBM Rechner hatte eben im Gegensatz zum Siemens Rechner kein Time-Sharing Modus. Das Testwerkzeug musste danach ausgerichtet werden.

Das nächste Modultestwerkzeug des Autors war von ganz anderer Natur. Es ging darum embedded Gerätesoftware für die Firma Kienzle zu testen. Die TU Karlsruhe hatte eine Programmiersprache für den Test von Mikroprozessoren entwickelt. Damit konnte man Objektcode für den jeweiligen Mikroprozessor generieren. Es fehlte jedoch die Möglichkeit den erzeugten Code in einer simulierten Umgebung zu testen. Der Autor wurde beauftragt diese Möglichkeit zu schaffen.

Die Mikroprogrammiersprache – SPM – wurde durch eine Skriptsprache ergänzt aus der einen Testtreiber und ein Teststub für jeden Port erzeugt wurden. Da das Zeitverhalten kritisch war dürfte der Code nicht instrumentiert werden. Die Zuweisung der Daten und die Testüberdeckung mussten auf anderer Weise bewerkstelligt werden, nämlich durch einen externen Testmonitor der den Test überwachte. Der Testtreiber sendete Eingangssignale an das Testobjekt und fing die Ausgangssignale ab. Während des Tests gab es keinerlei Unterbrechung. Erst nach dem Test wurden die Inhalte der Ausgangssignale mit den spezifizierten Ausgangswerten verglichen und Abweichungen registriert. Die Ablaufpfade wurden protokolliert und verglichen mit den erwarteten Pfaden. So konnte der Tester unregelmäßiges Verhalten der Software feststellen.

Das Kienzle Test und Dokumentationswerkzeug entstand in Zusammenarbeit mit der TU Karlsruhe. Es war das erste Realtime-embedded Softwaretestsystem in Deutschland und demonstrierte wie der Test in die

Programmiersprache eingebunden werden konnte. Auf der ACM Testkonferenz 1982 wurde über den Einsatz des Tools beim Test von Taximeter berichtet.

Das erste Integrationstestsystem in Deutschland wurde für die Lagerhaltung der Tornado Flugzeugteile in Augsburg konzipiert und getestet. Implementiert wurde sie in Budapest von den ungarischen Partnerinstituten des Autors. Dieses Testsystem – MBBTest - bestand aus einer Sammlung einzelner Testwerkzeugen die über Datenschnittstellen miteinander verbunden waren. Es gab einen Präprozessor um den PL/I Source Code für die Messung der Testüberdeckung zu instrumentieren. Ein Postprozessor wertete die Ergebnisse der Messung aus. Ein Datengenerator erzeugte aus dem Datenmodell ergänzt durch Datenregel die Testdatenbanken. Die Bewegungsdateien wurden aus einem Testskript mit Assertionen generiert. Ein Daten-Validator benutzten das gleiche Testskript um die Bildschirmausgaben mit dem Batch-Terminalsimulator zu prüfen. Ein Datei-Komparator verglich die VSAM Dateien vor dem Test mit denselben VSAM Dateien nach dem Test. In dem MBBTest waren viele der Komponente zu finden die heute zur Automation des Systemtests gehören. Es war ein Vorgriff auf die künftige Testautomation. Leider hat es in der deutschen Rüstungsindustrie einen Skandal ausgelöst das zur Verbannung des Autors führte.

Im März 1982 fand die erste deutsche Testkonferenz an der Bundeswehrhochschule in Neubiberg statt [ACM82]. Veranstalter war das German Chapter der ACM. Keynote Sprecher waren Prof. C. Ramamorthy von der Universität California, Berater des RXVP Projektes und Les Belady vom IBM Forschungslabour. Es gab mehrere Beiträge über den Test im deutschen Lande, unter anderem von der TU Karlsruhe die im Begriff war das RXVP Testsystem für Deutschland nachzubauen. Die Firma Kienzle berichtete über ihre Erfahrung mit dem Test von Mikroprozessoren und die jungen Unternehmer Heinz Bons und Rudolf van Megen stellten ihr TestCover Tool für die Messung der Testüberdeckung von COBOL Programmen vor. Von Siemens wurde über ihre Testmessungswerkzeuge berichtet. Vom SzKI Institut in Budapest wurde das von ihnen entwickelten Messsystem –SZ und von dem Szamok Institut wurde über den Einsatz des Softest Tools bei Bertelsmann berichtet. Diese Konferenz markierte den Beginn einer neuen Informatiktechnologie im deutschsprachigen Raum – der Softwaretest. Danach hat sich diese Technologie in der Industrie verbreitet und wurde zum Lehrfach an einigen Hochschulen allen voran die Universität Bremen wo Professor Franck die ersten Kurse zu diesem jungen Thema anbot. Die Gründung der GI-TAV Fachgruppe sollte erst 8 Jahre später folgen.

Literaturhinweise:

[ACM82] Sneed, H.M. Wiehle, H.R.: Software-Qualitätssicherung, Tagung I/1982, German Chapter of ACM, Neubiberg, Teubner Verlag, Stuttgart