

Chipkarten-Applikationen: Herausforderungen für die Software-Entwicklung

Von Markus Liegl



Die Software-Entwicklung von Chipkarten ist eine besondere: Der Speicherplatz ist begrenzt, Nachbesserungen sind nach Kartenausgabe nicht möglich und die Sicherheitsanforderungen sind enorm. Entwicklungszyklen und Testverfahren sind entsprechend aufwendig. Um die Karten dennoch möglichst fehlerfrei zu halten, setzt der Chipkartenhersteller Giesecke & Devrient modellbasierte Testverfahren ein, bei denen automatisch Zehntausende von Test-Cases erzeugt werden. Red.

Von der Öffentlichkeit eher weniger beachtet, haben auch Chipkarten in den letzten Jahren eine beachtliche technische Entwicklung vollzogen. Verfügten die ersten Karten nur über einen sehr kleinen Datenspeicher in Form eines Magnetstreifens, so sind ihre modernen Nachfolger mit Prozessor, Arbeitsspeicher, eigenem Betriebssystem und integrierter Kryptografie ausgestattet, vollständige Computer im Kleinstformat.

Sie erreichen heute eine Rechenleistung, die etwa der eines 15 Jahre alten Desktop-Computers entspricht. Die Anwendungsmöglichkeiten dieser Technologie sind denn auch in den letzten Jahren geradezu explodiert: Neben den klassischen Karten für den bargeldlosen Zahlungsver-

kehr werden Chipkarten heute für ganz unterschiedliche Aufgaben eingesetzt, zum Beispiel als Ausweiskarten für die Personenidentifizierung, als SIM-Karten für Mobiltelefone, als Gesundheitskarte oder als Zugangsschlüssel fürs Pay-TV. Zusätzliche Einsatzgebiete erschließt die neue Technologie der Java-Karten; hier können nicht nur die Hersteller, sondern auch die Kartenemittenten Applikationen auf die Karten laden, was deren Flexibilität enorm erhöht.

Spezifische Rahmenbedingungen

Alle diese unterschiedlichen Aufgaben werden natürlich durch Software bereitgestellt. Sie ist im Speicher der Karten implementiert und wird in Verbindung mit den jeweiligen Host-Systemen, also beispielsweise einem Handy oder einem Kassenterminal, aktiviert. Es versteht sich von selbst, dass die Applikationen für Chipkarten in höchstem Maße sicherheitskritisch sind: Sowohl beim Bezahlen als auch bei der Zugangskontrolle könnte der Einsatz falsch programmierter Karten fatale Folgen haben.

Zum Autor

Markus Liegl ist Leiter Basic Development Software-Tests bei der Giesecke & Devrient GmbH, München.

Die Entwicklung dieser Art von Software unterscheidet sich in einigen Punkten deutlich von „normaler“ Softwareentwicklung.

■ Das liegt zum einen an den technischen Voraussetzungen: Die Software muss immer mit dem Speicher auskommen, den der Chip-Hersteller zur Verfügung stellt.

■ Eine weitere Herausforderung für die Softwareentwicklung stellen die besonderen Sicherheitsanforderungen dar. Chipkarten müssen allen Arten von Angriffen standhalten, vom einfachen Abhören der Kommunikation zwischen Karte und Terminal bis zum Manipulieren von Inhalten.

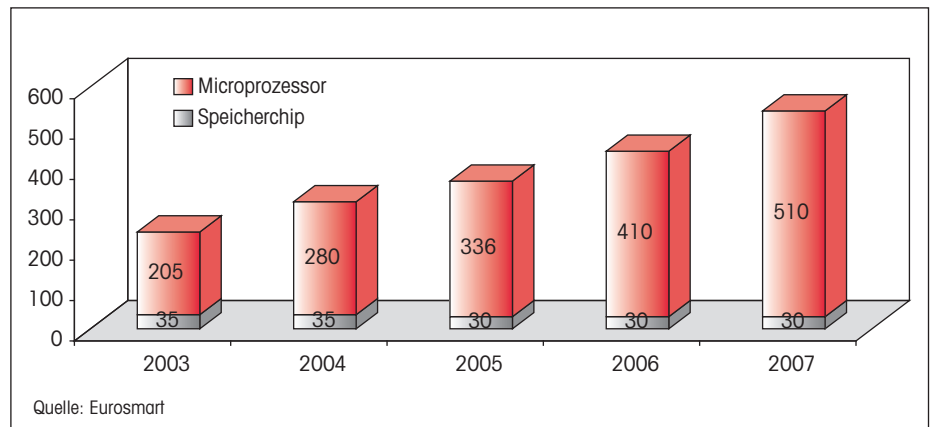
■ Schließlich können für Chipkarten-Applikationen auch nicht die üblichen Verfahren der Software-Pflege angewandt werden, denn sind die Karten einmal ausgegeben, können sie nicht nach einem halben Jahr zurückgerufen werden, um Patches oder Bugfixes einzuspielen. Außerdem befinden sich die Programme im ROM, lassen sich also nicht überschreiben. Nachbesserungen oder auch die üblichen Wartungszyklen sind hier ausgeschlossen. Auch nachträgliche Erweiterungen der Applikationen in Form von Updates sind für den Kartenhersteller kaum möglich, sodass auch die Spezifikation der Anforderungen sehr gründlich, genau und weit vorausschauend erfolgen muss.

Die Besonderheiten der Karten-Software stellen natürlich enorme Anforderungen an die Software-Qualität. Software, die das Haus verlässt, muss fehlerfrei sein. Die Entwicklungsdauer ist dementsprechend länger. In einem ausgefeilten Qualitätssicherungssystem wird im Rahmen des Test-Managements nicht nur nach Fehlern oder Ungenauigkeiten in der Programmierung gesucht, sondern auch geprüft, ob die Chips die Spezifikation tatsächlich einhalten oder ob es Fehler in den Tools gibt.

Organisatorisch sind die Softwareentwicklung und das Testen streng getrennt. Im Entwicklungsprozess beginnt das Testen frühzeitig, Test-Cases werden schon während des Entwicklungs-Zyklus aufgebaut. Die Test-Ingenieure leiten aus den Anforderungen geeignete Testfälle ab, die am Ende des Zyklus auf die Software angewandt werden. Solche Test-Läufe sind äußerst umfangreich, erstrecken sich über mehrere Tage und umfassen mehrere 10 000 Test-Cases.

Mit der Komplexität der Systeme ist auch die Komplexität der Tests in den letzten Jahren deutlich gestiegen. Wo es bis vor wenigen Jahren geschlossene Produkte mit Betriebssystem und Anwendung gab, die nach außen nur über eine Kommando-schnittstelle kommunizierten, gibt es heute Karten, die über eine Programmier-

Weltweite Auslieferung von Chipkarten im Bereich Finanzdienstleistungen/Kundenkarten 2003 bis 2005 (in Millionen)



schnittstelle verfügen, auf die der Kunde aufsetzen und ausführbaren Code laden kann. Diese Technologie stellt besonders hohe Anforderungen an die Test-Verfahren. Sie müssen alle Möglichkeiten vorhersehen und prüfen, wie sich das Produkt verhält. Und egal welche Programme auf die Karte geladen werden – die Sicherheit der Karte darf unter keinen Umständen davon tangiert werden.

Steigende Komplexität

Mit steigender Komplexität der Software steigt der Testaufwand überproportional, sodass für bestimmte Funktionalitäten manuelle Tests zu aufwendig werden. Bei steigender Komplexität und gleichem Auf-

wand lässt sich eine gleichbleibende Test-Coverage nur durch Automatisierung des Testens erreichen.

Noch vor wenigen Jahren waren automatisierte Tests für Chipkarten nicht notwendig. Heute geht es jedoch vor allem darum, den erreichten hohen Standard zu halten. Denn es wird immer schwieriger, sehr nahe an der 100-prozentigen Fehlerfreiheit zu bleiben. Die Qualität der Produkte darf nicht sinken, aber die Kosten der Qualitätssicherung müssen natürlich auch begrenzt bleiben.

Das Modell-basierte Testen, wie es von Leirios unterstützt wird, zählt hier zu den viel versprechenden Ansätzen. Damit können sehr viele Fälle und Fallkombinationen ausprobiert werden. Auch Erstellung von mehreren 10 000 Fällen sind kein Problem mehr. Manuell wäre das bei den heutigen Entwicklungszeiten nicht machbar.

Auch bei der Weiterentwicklung von Lösungen kann das modellbasierte Testen seine Vorzüge ausspielen: Sind Modelle bereits vorhanden, so lassen sich neue oder geänderte Anforderungen sehr leicht einarbeiten. Die Tester müssen nicht Hunderte von Tests durcharbeiten und prüfen, wo sich die Änderungen auswirken, sondern ändern das Modell und überlassen es dem Test-Generator, die entsprechend veränderten Test-Cases auszuarbeiten.

Marktüberblick kontaktlose Karten 2007* (in Millionen Karten)

