



Software kann fehlerfrei sein

Ein Ziel jeder Softwareentwicklung ist, fehlerfreie Programme zu erzeugen. Was die Businesslogik eines Programms betrifft (also die Funktionalität auf der Ebene der Anwendung) ist das noch *relativ* leicht zu verifizieren und sicherzustellen. Die gefährlichen und heimtückischen Fehler schleichen sich auf einer anderen Ebene ein: dem Umgang des Programmierers mit der Programmiersprache.

Die hier erzeugten „Bugs“ entstehen nicht dadurch, dass z.B. ein Algorithmus falsch abgebildet wurde. Sie resultieren vielmehr aus einer nicht ganz sachgerechten Verwendung der Möglichkeiten, die von der jeweiligen Sprache geboten werden. Das kann zum Beispiel dann passieren, wenn der Programmierer über nur ungenügende Kenntnisse der Arbeitsweise der Sprache verfügt, ihre Besonderheiten außer Acht lässt oder einfach nur unkonzentriert arbeitet.

Die Profis der InfraSoft haben sich die Aufgabe gestellt, einen Ansatzpunkt zum Vermeiden der häufigsten Fehler zu bieten. Das Ergebnis ist ein Dokument, in dem eine Zusammenstellung jener Bereiche zu finden ist, die am anfälligsten für fehlerhaften Code sind. Vor allem junge Programmierer werden die umfangreichen Hinweise zu schätzen wissen und können daraus lernen, wie Fehler in diesen kritischen Bereichen am Besten zu vermeiden sind. Interessierte finden den Text „Bug-freies Programmieren“ auf der InfraSoft Homepage unter www.infrasoft.at/service.



Die kritischen Bereiche

Wie Sie beim Lesen des Dokuments feststellen werden, gilt vor allem: Programmierfehler müssen nicht sein. Abgesehen vom richtigen Einsatz der Programmierer sowie der nötigen Sorgfalt und Disziplin des Einzelnen hilft vor allem eine genaue Kenntnis der kritischen Bereiche, in denen die meisten Fehler passieren. Zu diesen Bereichen zählen vor allem:



Softwarefehler sind in manchen Anwendungen einfach nicht leistbar. Eine genaue Kenntnis der kritischen Bereiche schützt vor gefährlichen Bugs.

- **Resource Management:** Unter Resources wird alles verstanden, was von einem Programm während der Laufzeit belegt und wieder freigegeben wird, also zum Beispiel Hauptspeicher, Festplattenspeicher oder Datenbankhandles – also generell alles, was ein Programm benötigt, um zu laufen, was es aber erst zur Laufzeit belegen kann. Die Herausforderung für den Programmierer liegt nun darin, im Rahmen des Resource Managements sicherzustellen, dass alle belegten Ressourcen auch wieder freigegeben werden. Dabei können je nach Programmiersprache (z.B. mit oder ohne Garbage Collector) die verschiedensten Bugs produziert werden.



InfraSoft als Partner

bietet Ihnen professionelle Unterstützung.

- **Exception Handling:** Exceptions sind Mechanismen moderner Programmiersprachen, um Fehler abzubilden. Sie funktionieren im Wesentlichen so, dass das Programm zu einer Stelle springt, an der ein Codestück zur Fehlerbehandlung steht. Das heißt, es wird ohne weitere Umwege der Errorhandling Code ausgeführt - der „normale“ Code wird in großen Zügen übersprungen. Daraus ergibt sich das Hauptproblem mit Exceptions: Wenn der Programmierer nicht damit rechnet, dass Teile seines Codes übersprungen werden, kann es zu Resource Leaks oder zu inkonsistenten ProgramMZuständen kommen.

- **Multithreading:** So bezeichnet man die parallele Verarbeitung von zwei oder mehreren Prozessen, die sich ein Datensegment teilen. Diese Programmieretechnik wurde lange Zeit völlig zu Unrecht als relativ leicht angesehen. Inzwischen weiß man, dass korrekte Multithreading Programmierung zu einer der schwierigsten Aufgaben überhaupt zählt. Für unerfahrene Programmierer lauern die Gefahren hier überall: Deadlocks, Race Conditions und Concurrency Situations sind mit Leichtigkeit zu produzieren. Sie führen zu Abstürzen, Fehlfunktionen und undefiniertem Programmverhalten.

- **Buffer Overflows:** Programmiersprachen bieten für die verschiedensten Zwecke Buffer sowie Funktionen zu deren Beschreibung an. Nicht alle dieser Funktionen schützen automatisch vor einem Überlaufen der Buffer. Die meisten der modernen Compiler bieten alternative Funktionen an, die auf die Länge des jeweiligen Buffers Rücksicht nehmen.

- **Wir setzen modernste Methoden ein**
Unsere Mitarbeiter verfügen über fundierte Kenntnisse in C++, Java, SQL, Notes und HTML/XML.

- **Wir stellen Know-how bereit**
Die InfraSoft-Teams haben langjährige Erfahrungen mit Datenbanken, Networking und Internet.

- **Wir schonen Ihr Budget**
Unsere Entwickler stehen *standby*. Sie bezahlen uns nur dann, wenn Sie uns brauchen.

- **Wir übernehmen Verantwortung**
Unsere Teams sind bereit, komplexe Probleme in Eigenregie zu lösen.

Vor Fehlern ist der Programmierer in diesem Bereich also nur geschützt, wenn er die verwendete Programmiersprache und die Arbeitsweise ihrer Funktionen genau genug kennt.

Die genaue Beschreibung dieser kritischen Bereiche (auf der InfraSoft Website kostenlos zum Download unter www.infrasoft.at/service) wurde von Harald Nowak, einem der erfahrensten Programmierer der InfraSoft erstellt. Der Autor steht unter anderem auch für individuelle C/C++ Trainings und Code-Reviews zur Verfügung. Mehr Informationen zu diesen Dienstleistungen erhalten Sie auf Anfrage unter info@infrasoft.at.



Die Profis von InfraSoft sind interessant für mich. Bitte kontaktieren Sie mich:

Vorname : _____
Nachname : _____
Firma : _____
Adresse : _____

Tel : _____
Handy : _____
Fax : _____
e-mail : _____
via http:// _____

