

... das aktuelle Stichwort ...

Laufzeitverlängerung ohne Restrisiko

Vergleichen wir doch einmal Software mit einem Atomkraftwerk. Wie bitte? Sie meinen, ein solcher Vergleich wäre völlig unsinnig und verbietet sich? In der Tat: Auf den ersten Blick gibt es wenige Gemeinsamkeiten. Fahren Sie zum Beispiel einmal ein Atomkraftwerk herunter. Ein abgeschaltetes Kernkraftwerk muss noch lange Zeit gekühlt werden. Endgültig aus ist es nie, denn die Brennstäbe strahlen ja immer weiter. Ein Programm hingegen beendet sich im Normalfall binnen Sekunden. Im schlimmsten Fall hängt es sich auf, endgültig. Und notfalls können Sie sogar den Stecker ziehen. Doch tun sich in anderer Hinsicht erstaunliche Parallelen zwischen Atomkraftwerken und guter Software auf: Man sieht, hört und riecht zum Beispiel nicht, was drinnen passiert. Beides sind komplexe Systeme, die von einem hohen Sicherheitsbedürfnis geprägt sind und eine umfassende Dokumentation verlangen. Genauso wie ein Kernkraftwerk besteht ein Softwareprodukt aus verschiedenen Komponenten. Bei der Verwirklichung gibt es von der Planung über Spezifikation, Erstellung, Inbetriebnahme und Tests mehrere Gewerke. Alles soll dann über einen möglichst langen Zeitraum fehlerfrei und ohne hohen Wartungsaufwand funktionieren. Auch ein Softwareprodukt hat ein unvermeidliches Restrisiko, denn ein größeres und dementsprechend komplexes Programm wird nie völlig fehlerfrei sein. Doch bei den Folgen einer Störung und bei der Entsorgung ist dann endgültig Schluss mit den Gemeinsamkeiten.

Sehen wir uns jedoch einmal die Langlebigkeit genauer an. Auch hier sind die Gemeinsamkeiten begrenzt, denn es gibt sicher nur wenige Softwareprodukte, welche die (durchschnittliche) Lebensdauer eines Atomkraftwerks erreichen. Doch es gibt sie, diese Ausnahmeprodukte, die meist in Nischen über Jahre hinweg ihre Funktionen zum Nutzen der Anwender bereitstellen. Dazu zählen Steuerungsprogramme, die während der gesamten Lebensdauer ihrer Geräte laufen, sowie viele individuell erstellte Adresskarteien, Lagerverwaltungen,



Volker Brase ist seit 25 Jahren Softwarearchitekt des Erlangener Informatikunternehmens Infolab. Er hebt die Erfolgsfaktoren für die Langlebigkeit von Software hervor.

Dokumentationssysteme oder Produktionssteuerungen, die vor allem in mittelständischen Betrieben zu finden sind. Diese Programme wurden häufig um einzelne Funktionen erweitert und an neue Abläufe angepasst, und sie sind in diesem langen Zeitraum zu einem unerlässlichen Bestandteil der Betriebsabläufe geworden. Das Personal kann sie wie im Schlaf bedienen. Ein Beispiel für langlebige Software ist das 1986 von Infolab für Siemens entwickelte Langzeitarchivierungs-Programm Autarch. Es erfüllt bis heute die Anforderung, Dateien über einen Zeitraum von bis zu 30 Jahren sicher zu archivieren. Dabei handelt es sich um Dateien von Stromlaufplänen der von Siemens und KWU errichteten Kraftwerke. Autarch hat im Laufe der Jahre mehrere Abteilungsumorganisationen und -umbenennungen überlebt und ist zweimal umgezogen. Während sein langjähriger Systembetreuer längst den Ruhestand genießt, arbeitet

Autarch immer noch zuverlässig. Durch die langen Laufzeiten erzielen solche Anwendungen sehr günstige Werte bei den Gesamtbetriebskosten (Total Cost of Ownership). Die Investitionskosten fallen über den langen Zeitraum kaum mehr ins Gewicht, und auch die Wartungskosten halten sich in Grenzen, denn eine stör anfällige Software wird kaum eine lange Lebensdauer erreichen.

Was macht eine Software, die über einen langen Zeitraum benutzt und weiterentwickelt werden kann, so erfolgreich? Erstens das Bausteinprinzip: Eine modulare Architektur mit definierten Schnittstellen erlaubt das Austauschen einzelner Komponenten. So können neue Techniken (auch ergänzend) eingebaut werden, ohne ein System als Ganzes zu beeinträchtigen. Logische Schichten erfüllen denselben Zweck: Wenn zum Beispiel Datenhaltung und Programmlogik getrennt sind, können neue Speichertechniken genutzt werden, ohne viel vom Code neu schreiben zu müssen. Je einfacher die Daten und Funktionen strukturiert sind, umso einfacher ist eine Weiterentwicklung. In der Vergangenheit musste oft aus Kapazitätsgründen vieles stark komprimiert oder abgekürzt abgelegt werden. Wenn die Beschreibungen dazu nicht mehr verfügbar sind, wird es teuer. Nur mit einer ordentlichen und aussagekräftigen Dokumentation ist eine kostengünstige Weiterentwicklung möglich. Letztlich steht hinter allem der dreifache Grundsatz: Qualität, Qualität, Qualität. Was ich heute nicht investiere, muss ich in der Zukunft unter dem Konto „Lehrgeld“ verbuchen. Nachhaltigkeit erziele ich nur, wenn ich heute schon an morgen denke. Wie den Atomkraftwerken droht heute vielen Softwareprodukten eine Abschaltung: Wegen drohender Inkompatibilität, zum Beispiel mit Windows 7, können viele Altanwendungen künftig nicht mehr oder nur noch mit beträchtlichem Verwaltungsaufwand weiterbetrieben werden. Der wichtigste Unterschied ist allerdings: Zu Atomkraftwerken gibt es eine Alternative, zu guter Software jedoch nicht.