© Peter Sommerlad

Professor Peter Sommerlad
Gründer und Leiter des Instituts für
Software an der HSR Hochschule
für Technik in Rapperswil, Schweiz

Zurück in die Zukunft der Software?

Als „Softwerker“ mit Leib und Seele möchte ich die Welt von schlechter Software befreien. Doch das wird vermutlich nicht gelingen, es sei denn, es gibt keine Hardware mehr. Der Weg ist das Ziel, so die Hoffnung. Wenn ich Embedded-Software-Leuten begegne, speziell im Automobilbereich, fühle ich mich oft wie auf einer Reise in die Vergangenheit. Die Branche beschäftigt sich heute noch mit Dingen, die vor Jahrzehnten aktuell waren.

Aber warum tut sie das? Nur wenige Embedded-Software-Ingenieure sind in Software Engineering ausgebildet. Viele stammen zudem aus anderen Disziplinen. Ein weiterer Grund für eine, ich nenne sie mal „konservative“ Haltung ist die limitierte Hardware. Erfahrung anderer auf die diesem Gebiet zu ignorieren und Fehler zu wiederholen ist allerdings gefährlich – speziell heute, wo 32-bit-Prozessoren und Megabytes 8 bit und wenige Kilobytes ablösen.

Ich bemerke: Die Automobilbranche setzt sich nur widerwillig mit gutem Softwaredesign und moderner Softwaretechnik auseinander – beispielsweise mit der Kapselung mit klaren Schnittstellen, „Separation of Concerns“, mit Parametrisierung, effizienter Abstraktion, aktuellen Programmiersprachen, mit Compilern und IDEs sowie dem Prinzip der „Simplicity“. Letzteres wird häufig fehlverstanden als Ausrede für die Nutzung vermeintlich einfacher Sprachen, die anstatt hochwertiger Abstraktionsmechanismen nur primitive Konstrukte bieten.

Softwareentwicklung mit solchen Sprachen resultiert in unnötig aufgeblähtem, unverständlichem Code. Einfachheit von Software bedeutet aber: Nutzen von Abstraktionen und Schnittstellen zur Interaktion zwischen Komponenten für weniger und besser wartbaren Code. Für diese Designvereinfachung muss bestehender Code umgebaut und aufgeräumt werden. Und genau das wird vermieden. Vorteile werden dabei unterschätzt und jegliche Änderungen werden als Gefahr eingestuft.

Eine letzte Mahnung möchte ich in Bezug auf qualitätsorientierte Kostenrechnung geben: Viele Unternehmen investieren sowohl in Prozesse mit hohem Reifegrad als auch in agile Prozesse, die übrigens oft missverstanden werden. Doch diese Investitionen kompensieren nicht die Folgekosten, die durch schlechte Software entstehen. Billigere Herstellung und Produkte mit geringerer Qualität mögen in anderen Bereichen für höhere Profite sorgen. Bei billiger Software funktioniert das nicht, außer beim Dienstleister, der nach Aufwand abrechnet.

Kommen Sie „zurück in die Zukunft“. Schließen Sie zum „Jetzt“ im Software Engineering auf. Fordern Sie von Ihren Partnern die Implementierung aktueller Standards, und vermeiden Sie Fehler, die andere schon vor Jahrzehnten gemacht haben. Nutzen Sie moderne Sprachen. Und geben Sie Ihren Ingenieuren Zeit zum Lernen. Wandel ist nun mal anstrengend, der Mut zur Veränderung zahlt sich aber durch einfachere Lösungen für schwierigere Probleme aus.