

Komplexität und Modell

Software als Rohstoff der Industrie 4.0. Ein Pfeiler der Zukunftstauglichkeit deutscher Ingenieurskunst ist das Software-Engineering. Auch hier vollzieht sich ein Wandel, denn das Thema Komplexität hat einen Paradigmenwechsel notwendig gemacht: vom Programmieren zur Modellieren.



Die strategische Entscheidung, diesen Paradigmenwechsel beim Software-Engineering auch im eigenen Unternehmen zu vollziehen, wird für viele Unternehmen existenziell sein. Doch häufig wissen die Manager der obersten Führungsebene wenig von den Softwareentwicklungsmethoden der Zukunft. Sie überschätzen die Kosten für die Einführung neuer Tools und Methoden und unterschätzen den Return-on-Invest.

Dabei geht es um nicht weniger als das sichere Wissen, für die nächste Generation des Software-Engineerings – und für die Industrie 4.0 – zukunftssicher gewappnet zu sein. Andreas Willert, CEO bei Willert Software

Tools, diskutiert die Herausforderungen, welche Ingenieure, Softwareentwickler und Manager künftig gemeinsam meistern müssen.

eli: Die Komplexität und Dynamik elektronischer Systeme wächst; sie lassen sich nicht mehr ohne Weiteres durchschauen. Welche Strategien gibt es, um sie dennoch zu beherrschen?

Willert: Eine der am häufigsten eingesetzten Methoden, um ein kompliziertes System zu verstehen, ist die Zerteilung. Teile und herrsche – dieses Instrument ist bis heute erfolgreich, bei komplexen Systemen jedoch mit immer weniger Erfolg.

eli: Kompliziert ist also nicht gleich komplex?

Willert: Nein, und darin liegt das Problem. Die Zerteilung wird so lange fortgeführt, bis die Teilsysteme in ihrer Kompliziertheit beherrschbar erscheinen. Allerdings erzeugt das Zerteilen Schnittstellen, und zwar mit steigender Zahl der Teilsysteme überproportional viele, sodass auf der Ebene dieser Schnittstellen eine neue Kompliziertheit entsteht. Wir bilden mit dieser Methode nur eine von vielen Ebenen des Verstehens ab. In realen Softwareprojekten sind aber immer mehr Ebenen – wie Zeit, Versionen, Varianten und Betriebsmodi – zu berücksichtigen. Hier entsteht eine Komplexität, die sich mit der Teile-und-Herrsche-Methode nicht mehr vollständig verstehen lässt.

eli: Und das ist ein schwerwiegendes Problem für Softwareentwickler.

Willert: So ist es. Der Grad der Vernetzung über mehrere Ebenen macht es nahezu unmöglich, die dynamische Auswirkung einer Änderung in einer statischen, eindimensionalen Betrachtung vorherzusagen. Genau das möchten wir aber gern: beherrschbare, vorhersagbare Systeme, auch und gerade bei der Software. Die herkömmliche Entwicklung von Software auf der Basis von Hochsprachen, dazu noch in verteilten Teams, erreicht die Leistungsgrenzen unserer Gehirne. Immer häufiger kommen Systeme in Situationen, in denen die Kombination von Zusammenhängen und deren Auswirkungen bei der Entwick-

KONTAKT

Willert Software Tools GmbH,
Hannoversche Straße 21,
31675 Bückeburg,
Tel. 05722 967860,
Fax 05722 967880,
www.willert.de

lung nicht mehr zu überblicken sind. Fachlich sprechen wir hier von einer Emergenz – von unvorhersehbaren Zuständen, die im schlimmsten Fall zu gravierenden Fehlfunktionen führen. Ein tragisches Beispiel war der Absturz eines A400-Transportflugzeugs im Mai 2015, den Softwarefehler verursacht hatten.

eli: Komplexe Software wird wegen ihrer Funktionen gebraucht, ihre Komplexität behindert jedoch gleichzeitig das Verstehen. Gibt es einen Weg aus diesem Dilemma?

Willert: Den gibt es durchaus, und es stehen uns drei Mechanismen dafür zur Verfügung: Musterbildung, vertragsbasierte Programmierung und Abstraktion. Alle drei hängen eng miteinander zusammen und bedingen sich gegenseitig.

eli: Würden Sie das bitte erklären?

Willert: Die Musterbildung bringt Routinen in die Softwareentwicklung. Sind die Muster einmal gelernt, funktionieren sie wie Blaupausen. Die vertragsbasierte Programmierung sichert das reibungslose Zusammenspiel einzelner Programmmodule anhand formaler Verträge zur Verwendung

STECKBRIEF

Dipl.-Ing. (FH) Andreas Willert ist seit 1989 im Software-Engineering tätig. Er begann im Umfeld der C-Programmierung und beschäftigt sich seit nunmehr zehn Jahren mit der Modellierung, dem Requirements Engineering und der Qualitätssicherung auf dem Gebiet der Embedded-Systeme. Neben seiner Tätigkeit als Geschäftsführer bei Willert Software Tools gibt er seine Erfahrungen als Autor, Trainer und Referent weiter.



die Absicherung von Theorien nicht nur sehr preiswert, sondern auch sehr früh im Entwicklungsprozess möglich. Das verringert neben den Kosten vor allem das Risiko von Fehlentwicklungen und verkürzt die Entwicklungszeit.

eli: Wie wir es aus der klassischen Hardwareentwicklung kennen.

Willert: Aber dort werden Modelle ausschließlich bis zur Serienreife genutzt. Im Software-Engineering sind Modelle noch viel vorteilhafter: Denn es entstehen keine Materialkosten, Software ist frei von Verschleiß, und sie wird in Versuchen

eli: Was bedeutet die Einführung von Modellierung für die Entwicklungsabteilungen?

Willert: Sie bedeutet ein Umdenken – in allen Prozessen der Softwareentwicklung, aber auch in der Zusammenarbeit im Team. Ein spannender Weg, der ohne die Bereitschaft aller zur Veränderung nicht gegangen werden kann. Mit dem Wechsel zur Softwaremodellierung entstehen neue Arbeitsmodi auf einem aktuellen, zukunftsweisenden Kompetenzniveau.

eli: Die Entscheidung für einen solchen Paradigmenwechsel liegt aber nicht in der Verantwortung der Entwickler, sondern eher beim Management.

Willert: Ingenieure können aber das Management darauf hinweisen, dass die gegebenen Arbeitsmittel der Komplexität nicht mehr gewachsen sind. Und sie können Empfehlungen für notwendige Schritte aussprechen. Ist dann die Entscheidung pro Modellierung gefallen, sind Pilotprojekte sinnvoll. Die Kosten für die Einführung dieses neuen Vorgehens einschließlich der Werkzeuge und Schulung entsprechen, bei solider Ausführung, in etwa denen von zwei Mann-Monaten. Kunden, die diesen Schritt vor fünf bis zehn Jahren konsequent gemacht haben, berichten uns heute von Effizienzsteigerungen auf das Zwei- bis Dreifache.

Danke für das Gespräch.

ml

Das Interview führte Annette Menzel, Flowmedia.

Online-Service

Andreas Willert: Sieben Thesen für eine solide modellgetriebene Softwareentwicklung

www.elektronik-informationen.de/44067

„Es entsteht eine Komplexität, die sich anhand der Teile-und-Herrsche-Methode nicht mehr verstehen lässt

Andreas Willert

von Schnittstellen. Und Abstraktionsmechanismen beschreiben anhand von Modellen komplexe Zusammenhänge auf eine einfachere Art und Weise. Die Sprache, die wir dazu nutzen heißt UML: Unified Modeling Language. Aus solchen Modellen, in UML beschrieben, lässt sich dann der benötigte Code generieren. Im Software-Engineering ist ein Paradigmenwechsel von der Programmierung zur Modellierung nötig, um mit der Komplexität adäquat umgehen zu können. Dies ist ein entscheidender Schritt in der Bewältigung von Komplexität, weil das Abstraktionsniveau eines Modells das Verstehen erleichtert.

eli: Meinen Sie ein Modell im klassischen Sinn?

Willert: In gewisser Weise, ja. Modelle werden im Engineering seit Jahrzehnten sehr erfolgreich genutzt. Anhand von Modellen ist

nicht zerstört wie Materie bei einem Crashtest. Im Software-Engineering werden Modelle für die Entwicklung des Serienprodukts verwendet. Das setzt voraus, dass die Qualitätsattribute des Modells denen des Seriencodes entsprechen. Programmiersprachen wie UML ermöglichen es, auf hohem Abstraktionsniveau zu arbeiten und dabei hochwertigen Produktionscode aus den Modellen zu erzeugen. Die Modellierung mithilfe moderner Modellierungssprachen und geeigneter Werkzeugen ist Real-Time-Engineering in der Softwareentwicklung. Mit Modellen ist zum Beispiel die Absicherung von Theorien nicht nur sehr viel preiswerter, sondern auch sehr viel früher im Entwicklungsprozess möglich als ohne sie. Das verringert neben den Kosten vor allem das Risiko fehlerhafter Entwicklungen und verkürzt die Entwicklungszeit.