

Architekturzentrierte Softwaretechnik

Christiane Floyd

Arbeitsbereich Softwaretechnik, Department für Informatik
Universität Hamburg
Vogt-Kölln-Straße 30
22607 Hamburg
floyd@informatik.uni-hamburg.de

Kurzfassung

Die Konferenz SE'07 zeichnet sich unter anderem dadurch aus, dass im Programmkomitee sämtliche deutsche Forschungs- und Lehrinstitute vertreten sind, die sich mit Softwaretechnik bzw. Software Engineering befassen. Das gibt Gelegenheit zum Austausch zwischen verschiedenen Denkschulen. Für mich ist es zugleich eine Gelegenheit zur Bestandsaufnahme: Ich habe 1978 die erste Professur für Softwaretechnik angetreten und stehe nun kurz vor der Emeritierung. Ich habe eine Fachauffassung entwickelt, die durch Orientierung auf die Belange von Nutzern und Nutzerinnen gekennzeichnet ist. Eine ernsthafte Orientierung auf die Angelegenheiten der Nutzung ist aber nur dann möglich, wenn wir den sich allmählich entfaltenden Anforderungen der verschiedenen Beteiligten entsprechen und dabei Qualität gewährleisten können – dazu bildet die Software-Architektur die wichtigste Voraussetzung. Deshalb habe ich stets der Software-Architektur eine hohe Aufmerksamkeit geschenkt. Diese Anliegen teilt die gesamte Softwaretechnik-Denkschule an der Universität Hamburg.

In den Anfangsjahren meiner Fachvertretung wurde in der Softwaretechnik von „Architektur“ noch wenig gesprochen. Die grundlegenden Veröffentlichungen handelten über Entwurf (design), Zerlegung (decomposition), Modularisierung oder Programmierung im Großen; entsprechende Kapitel oder Untertitel finden sich in den ersten Lehrbüchern. Ende der 80er Jahre war der Begriff Softwarearchitektur schon eingebürgert, die Schwerpunktsetzung auf Architektur wird der Carnegie-Mellon-Schule aus den 90er Jahren zugeschrieben, die Wortschöpfung „architekturzentrierte Softwareentwicklung“ ist im Zusammenhang mit dem Unified Process entstanden. Seither nimmt der Fokus auf Architektur ständig zu. Jetzt steht sogar ein neues Berufsbild „Software-Architekt“ bzw. „Software-Architektin“ zur Diskussion.

Im Anschluss an die gängige Literatur verstehe ich im folgenden unter Software-Architektur die Zerlegung eines Softwaresystems in Einheiten oder Komponenten und die Spezifikation ihres Zusammenwirkens über festgelegte Verbindungen oder Schnittstellen. Ich beschränke mich dabei im Wesentlichen auf die statische Sicht von Architektur, die im Kleinen meist mit Mitteln der Programmiersprache ausgedrückt wird, im Großen auf Begriffe wie Subsysteme oder Schichten zurückgreift.

In diesem Vortrag geht es nicht primär um ein neues Rollenverständnis und auch nicht um den Stellenwert der spezifischen Architektur des entstehenden Systems im Rahmen eines Softwareprojektes. Im Mittelpunkt steht vielmehr ein tief greifender Wandel, der sich in den letzten Jahrzehnten in der Softwaretechnik vollzogen hat und noch vollzieht. War ursprünglich das Softwareprojekt die alles organisierende Betrachtungseinheit, um die einzelne softwaretechnische Themen angeordnet wurden, so nimmt heute zunehmend die Architektur die Stellung eines zentralen Gesamtanliegens ein. Dies eröffnet die Möglichkeit, alles andere um sie zu gruppieren. Um auszuloten, wie sinnvoll dies ist, wird das Thema Architektur in der Softwaretechnik in seiner Entwicklung betrachtet.

Die maßgeblichen Prinzipien der Software-Architektur – wie Schichtung, Information Hiding, Kapselung, Kohäsion und Kopplung – wurden bereits um 1970 publiziert, seither kontinuierlich vertreten und sind auch heute noch aktuell. Es gab auch schon sehr früh Muster-, Referenz- und Modellarchitekturen – die jedoch orientierten sich am damaligen Stand der Kunst und wurden später durch andere verdrängt. Gerungen wurde (und wird immer noch) um das Verhältnis von Architektur einerseits zum Programm und andererseits zum Modell. In einer „architekturzentrierten Softwaretechnik“ wird die Architektur als Eckstein der gesamten Softwareentwicklung gesehen.

Grundlegend geändert hat sich das Paradigma der Softwaretechnik und damit auch die Einordnung der Architektur. Das Ausgangsparadigma war zweifellos die Entwicklung eines Produktes nach einem Phasenmodell, in dem „Entwurf“ oder etwas Ähnliches als Phase vorgesehen war – ein zeitlich abgegrenzter Entwicklungsschritt mit vorgeschriebenen Ergebnissen. Der „Entwurf“ bezog sich damals auf das gesamte Produkt, dieses Produkt war von anderen Produkten weitgehend unabhängig. Die Anforderungen an das Produkt sollten vorher festgelegt werden, die Spezifikation des Entwurfs wurde als Grundlage für die arbeitsteilige Programmierung verstanden. So weit, so gut.

Seither haben wir mehrere verflochtene Entwicklungslinien miterlebt und mitgestaltet, die dieses Paradigma aufgelöst haben. Mitte der Neunziger Jahre wurde rückblickend eine Folge von Jahrzehnten konstatiert: das Jahrzehnt des Produktes, gefolgt vom Jahrzehnt des Prozesses und dem Jahrzehnt der Produktlandschaften. Wo stehen wir jetzt? Vielleicht im Jahrzehnt der verflochtenen Einzelprozesse, in denen miteinander auf vielfältige Weise zusammenhängende Produkte (weiter-)entwickelt, verändert und aufeinander abgestimmt werden. Schauen wir uns die Hauptlinien dieser Entwicklung etwas näher an.

Zum einen die Entwicklungslinie von linearen Vorgehensmodellen hin zu zyklischen (inkrementellen und iterativen) mit Prototyping und Ausbaustufen bis hin zu agilen Methoden. Dabei wurde der zunächst empfohlene und später als „up front design“ kritisierte Gesamtentwurf als Falle erkannt und – je nach Denkschule – durch mehr oder weniger emergente Vorgehensweisen ersetzt.

Zum anderen die Weiterentwicklung der Programmiermethodik. Die Orientierung auf Zerlegung markierte den Übergang von der strukturierten zur modularen Konstruktionstechnik. Der Top-Down-Anspruch wurde fallen gelassen und der Gesamtentwurf in Ausbaustufen aufgelöst. Die Objektorientierte Konstruktionstechnik mit dem Bottom-Up-Ansatz kennt allmählich wachsende Systeme und unterstützt komponentenbasierten Entwurf sowie die Verwendung von Rahmenwerken.

Schließlich hat sich die Aufmerksamkeit aus guten Gründen zunehmend von der Entwicklung neuer Software zur Erschließung, Bewahrung und Weiterentwicklung existierender Software im Rahmen von Refactoring, Re-Engineering, Portierung und Migration verlagert. Diese Bereiche wurden vielfach vom Rest der Softwaretechnik getrennt und im Rahmen der Wartung thematisiert, wobei manch ein Rad aufs Neue erfunden werden musste.

All dies geschah vor dem Hintergrund

- der technischen Entwicklung – mit den Übergängen von Stapelverarbeitung zu Interaktion, von Großrechnerlösungen zu vernetzten PCs bis hin zum Web – die die postulierte weitgehende Unabhängigkeit von Softwareprodukten in Frage stellte und ihr Zusammenspiel mit einer Fülle von anderen Produkten und Dienstleistungen mit sich brachte,
- sowie der Anwendungsentwicklung, bei der sich die Softwaretechnik domänenorientiert immer mehr ausdifferenzierte, die Wiederverwendung von Software wesentlich wurde und applikationsübergreifende Standards zu berücksichtigen waren.

Im Zuge dieser Entwicklungslinien wurde auch das Verständnis von Software-Architektur in der Softwaretechnik immer stärker ausgearbeitet und geht in mancher Hinsicht über das in anderen Bereichen etablierte Verständnis von Architektur hinaus.

- Software-Architektur beschreibt die Zerlegung in Einheiten innerhalb eines Softwaresystems (im Großen wie im Kleinen), aber auch das Zusammenspiel von Software mit anderen Komponenten der technischen Umgebung sowie mit anderen Produkten, Anwendungen oder Dienstleistungen, die auf die eine oder andere Weise als Gesamtheit angesehen werden.
- Software-Architektur hat eine zeitliche Dimension: Bei der (Erst-)entwicklung von Systemen entsteht (emergiert) sie im Lauf der Zeit, für die Weiterentwicklung von Software ist sie der zentrale Bezugsrahmen. So lassen sich verschiedene Stadien der Architekturentwicklung sowie zu jedem Zeitpunkt der Übergang von einer Ist-Architektur zu einer Soll-Architektur unterscheiden.

Aus der Fülle von praktischen Erfahrungen, theoretischen Konzepten, methodischen Ansätzen und technischen Konstruktionsanleitungen zu Software-Architektur(en) ist inzwischen Architektur als ausgereifter Teilbereich der Softwaretechnik entstanden.

Das Thema Architektur in den Mittelpunkt zu stellen, erlaubt eine Sichtweise auf Software, die die zeitliche Beschränkung auf das einzelne Projekt übersteigt, die Entwicklung und die verschiedenen Formen der Weiterentwicklung von Software in einem gemeinsamen Bezugsrahmen gleichrangig berücksichtigt und das einzelne Produkt im Zusammenhang mit anderen zu sehen gestattet.

Eine architekturzentrierte Softwaretechnik reicht von „harten“ Engineering-Ansätzen, die auf Messungen beruhen, bis zu „weichen“ Ansätzen der Teambildung, bei denen durch Zusammenarbeit und Vernetzung ein gemeinsames Verständnis der Architektur gefördert wird. Sie greift formale Ansätze in praktikabler Weise auf und schafft eine Sprachebene, um funktionelle wie nicht funktionelle Anforderungen mit Entwurfsentscheidungen, die zur Bildung von Architektureinheiten führen, in Verbindung zu bringen. Sie sieht in der Architektur bzw. in ihrer zeitlichen Entwicklung das ordnende Element für Softwarekonstruktion, Programmierung und Test, Produktverwaltung und -konfiguration. Sie fördert das Verständnis existierender Architekturen und Architekturstile als Ausgangspunkt für die Entwicklung bzw. Weiterentwicklung von Software.

So werden die fachlichen Voraussetzungen geschaffen, um gebrauchstaugliche, qualitativ hochwertige Software zu entwickeln. Um das in der Praxis auch zu gewährleisten, bedarf es einer kontinuierlich eingeübten und reflektierten Disziplin von Einzelnen und von Entwicklungsteams. Ob das nun die Rolle von Software-Architekten und -Architektinnen ausmacht oder auch für Software-Ingenieure und -Ingenieurinnen relevant sein kann, will ich nicht entscheiden.

Danksagung

Die hier geäußerten Auffassungen wurden zu einem Gutteil durch das abgeschlossene Promotionsvorhaben von Wolf-Gideon Bleek sowie durch die noch laufenden Promotionsvorhaben von Martti Jeenicke und Carola Lilienthal inspiriert. Dazu kommt die wertvolle kontinuierliche Diskussion mit Heinz Züllighoven, Axel Schmolitzky und den anderen Mitgliedern unseres Arbeitsbereichs.