mehr zum thema:
www.lukehohmann.com

SOFTWAREARCHITEKTUR UND MARKETING – ZUSAMMENPRALL ZWEIER WELTEN?

für sie gelesen von:

Kerstin Dittert (E-Mail: kerstin.dittert@oocon.de), freie Beraterin mit den Schwerpunkten Softwarearchitektur, OO-Methodik, J2EE und Projektmanagement.

Wie schön könnte die Welt der Softwareentwicklung sein, wenn die von uns erstellten Produkte nicht irgendwann (möglichst profitabel) an den Kunden gebracht werden müssten. Wir könnten frei konfigurierbare, hochgradig portable Systeme entwerfen und erstellen, die stets auf den neuesten Technologien basierten. Jeden Morgen würden wir uns einer neuen technischen Herausforderung stellen und sie durch unsere brillanten Lösungsstrategien bezwingen. Sollten irgendwann neuere oder bessere Basistechnologien zur Verfügung stehen, so würden wir unser tolles System samt Schnittstellen etc. einfach in die Tonne werfen, die Ärmel hochkrempeln und mit einem schwungvollen Redesign beginnen. Wie spannend und abwechslungsreich wäre dann die Softwareentwicklung!

Ein kleines Manko hat dieser Ansatz jedoch: Wahrscheinlich wäre kein Kunde bereit, für ein derartiges System auch nur einen Cent auszugeben. Unsere Marketing-Abteilung würde verzweifeln. Bald wären wir unseren schönen Job los, weil die von uns erstellten Systeme einfach keinen langfristigen Mehrwert bieten würden.

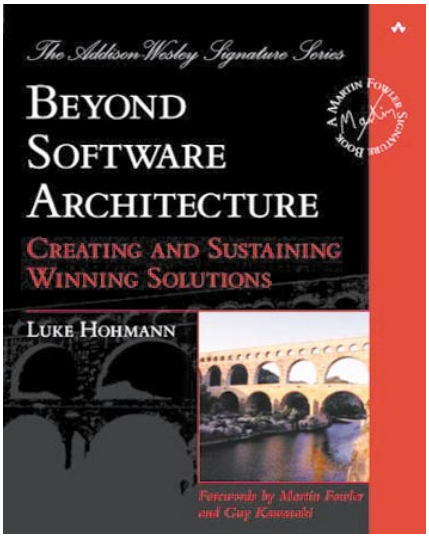
Genau an dieser Stelle setzt Luke Hohmanns Buch „Beyond Software-Architecture: creating and sustaining winning solutions" an. Der Autor betrachtet die Softwarearchitektur aus dem Blickwinkel des Produktmanagers und beleuchtet das Spannungsfeld zwischen Marketing und technischem Entwicklungsteam. Er beschreibt die Schlüsselfaktoren für Softwarearchitekturen, die sowohl *technisch* als auch *wirtschaftlich* erfolgreich sind. Luke Hohmann geht viele Fragen sehr pragmatisch an, was sicherlich ein Resultat seiner langjährigen Erfahrung als Projekt- und Produktmanager ist. Mit zahlreichen Praxis-Beispielen veranschaulicht er seine Strategien im Kontext konkreter Projekte. Ganz eilige Leser verschaffen sich bereits mit diesen Beispielen einen Überblick über wichtige Fragestellungen und Lösungsansätze.

Das Buch spricht Projektmanager, Softwarearchitekten, QS-Verantwortliche und Entwickler in Softwareentwicklungsprojekten an, die

- entweder *Standardsoftware*-Produkte für einen externen Markt oder
- größere und langlebige *Individuallösungen* für einzelne Unternehmen

erstellen.

Hohmann betrachtet die Interaktionen zwischen dem Geschäftsmodell und der Architektur eines Softwaresystems. Dieser zentrale Aspekt zieht sich wie ein roter Faden durch das gesamte Buch. Nach



Auffassung des Autors gibt es weitreichende Verflechtungen zwischen der Softwarearchitektur und dem Geschäftsmodell einer Anwendung. Hierbei kann sowohl das Geschäftsmodell spezielle Anforderungen an die Architektur definieren als auch umgekehrt. **Kasten 1** nennt Beispiele aus Hohmanns Buch für derartige Interaktionen.

Im Einführungskapitel über Softwarearchitektur unterscheidet der Autor „gute" von „schlechten" Architekturen und verweist auf bewährte Lösungsansätze wie Entwurfsmuster, Schichtenbildung etc. Der erfahrene Softwarearchitekt kann dieses Kapitel getrost überspringen, da er hier nichts Neues entdecken wird.

Die Sichtweise des Produktmanagers

In den folgenden Kapiteln konzentriert sich Hohmann auf das Marketing standardisierter Softwareprodukte. Er behandelt Themen wie

- Produktmanagement,
- Marksegmentierung,
- Geschäftspläne,
- Geschäftsmodelle,
- Lizenzmodelle und
- Schutzmarken.

Der Autor erwähnt unter anderem eine Pattern-Sprache für Produktmanagement, die er in einem Anhang detailliert beschreibt.

Der Zusammenhang zur Softwarearchitektur ist in diesen Kapiteln eher lose und sporadisch. Wer kommerzielle Softwaresysteme für einzelne Unternehmen erstellt, wird hier nur wenig Anregungen zur Verbesserung seiner Softwarearchitekturen finden. Demjenigen, der mit der Konzeption von Standardprodukten beschäftigt ist, bietet Hohmann aber interessante Einblicke in die Betrachtungsweise eines Produktmanagers. Missverständnisse und Vorurteile können unter Umständen abgebaut werden und zu einer Verbesserung des gesamten Softwareentwicklungsprozesses führen.

Kleine Ursache – große Wirkung

Im hinteren Teil des Buches steht die Technik stärker im Vordergrund. Hier geht es um grundlegende Architekturentscheidungen und ihre direkten Auswirkungen auf den Kundennutzen. Unter dem Aspekt der Wirtschaftlichkeit für Anwender und Softwarehersteller wendet sich der Autor nun folgenden Themen zu:

- Portabilität,
- Softwareverteilung,
- Integration und Erweiterung,
- Benutzbarkeit,

Interaktionen zwischen Softwarearchitektur und Geschäftsmodell

Beispiel 1: Geschäftsmodell –> Softwarearchitektur
Im Geschäftsmodell ist eine Lizenzierung auf Transaktionsbasis vorgesehen (z. B. für Web-Services, die im Internet angeboten werden). Daraus resultieren für die Softwarearchitektur eine Reihe von Anforderungen, z. B. an das Transaktionsmodell, die Session-Verwaltung sowie die Transaktions-Sicherheit.

Beispiel 2: Softwarearchitektur –> Geschäftsmodell
Das Geschäftsmodell unserer Anwendung sieht jährlich abzurechnende Firmenlizenzen vor.
Unser System verwendet eine Komponente eines dritten Herstellers. Dessen Lizenzmodell sieht ausschließlich die Abrechnung nach „concurrent user" vor.
In Verhandlungen wird versucht, mit dem Hersteller der Komponente einen speziellen Lizenzvertrag auszuhandeln, der mit unserem Geschäftsmodell vereinbar ist.
Sollten diese Verhandlungen scheitern, so müssen wir entweder einen Ersatz für die Komponente finden oder wir müssen unser eigenes Geschäftsmodell anpassen.

Kasten 1: Beispiel aus dem Buch „Beyond Software Architecture"

- Installation,
- Upgrade,
- Konfiguration,
- Logging,
- Release-Management und
- Sicherheit.

Die Lösungsansätze sind konkret und durch die Beispiele leicht nachvollziehbar. Diese Kapitel liefern wertvolle Hinweise, wie sich der Kundennutzen steigern lässt und welche architektur-relevanten Faktoren dabei zu beachten sind. Hier kommen auch alle Leser auf ihre Kosten, deren Tagesgeschäft nichts mit der Erstellung von Standardsoftware zu tun hat.

In dieser zweiten Buchhälfte haben mir die Kapitel über *Installation, Integration und Erweiterung* sowie *Upgrade* besonders gut gefallen. Die Betrachtungsweise ist vielfach erfrischend neu und die Empfehlungen sind in der Regel einfach und praktikabel.

Beispielsweise kann eine schlechte Installationsprozedur den Anwender sehr schnell und nachhaltig gegen eine Software einnehmen. Wer will sich schon mit Datenmüll nach fehlgeschlagenen Installationsroutinen herumärgern oder gar von der Installationsprozedur bedroht werden nach dem Motto: „Wenn Sie während der Installation nicht alle anderen Anwendungen schließen, könnte Ihr System unwiederbringlich kaputt gehen!" In der Regel sind solche „Drohungen", wie der Autor treffend formuliert, technisch unnötig und damit inakzeptabel.
Hohmanns Blickwinkel offenbart sich ebenfalls sehr gut in seinen Betrachtungen zum Thema Programmierschnittstellen. Wer hat sich nicht schon schwarz geärgert über Softwarekomponenten, die alle drei Monate mit einer neuen Programmierschnittstelle aufwarten – und schlimmstenfalls weder dokumentiert noch aufwärtskompatibel sind! Genauso geht es den Anwendern der *von uns* erstellten Softwaresysteme: Unsere Anwendungen sind in der Regel keine Insel, sondern sie werden in die komplexe Systemlandschaft unserer Kunden integriert. Schnittstellenänderungen können zu erheblichen Folgekosten führen und unsere Anwender unter Umständen von fachlich dringend benötigten Upgrades abhalten.

Bezogen auf das *Release-Management* fand ich Hohmanns Ansätze zur Analyse des Kundennutzen einzelner Systemfunktionen besonders interessant. Gerade bei sehr langlebigen Softwaresystemen kann es passieren, dass Features lange Jahre mitgeschleppt werden, ohne dass sie noch einen nennenswerten Nutzen für irgendeinen Anwender bieten. Der Autor beschreibt verschiedene Heuristiken, mit denen ermittelt werden kann, ob und in welchen Umfang bestimmte Systemteile genutzt werden. Durch diese Informationen kann ein Softwaresystem auch über längere Zeiträume immer wieder an die *wirklichen* Kundenbedürfnisse angepasst und erweitert werden, ohne irgendwann zur unüberschaubaren „Feature-Hölle" zu mutieren.

Look&Feel
Die Struktur des Buches hat mir ausgesprochen gut gefallen: Die Unterteilung in sechzehn Kapitel liefert handliche Einheiten, die zum größten Teil auch unabhängig voneinander gelesen werden können. Den Abschluss jedes Kapitels bilden

- eine Kurzzusammenfassung in Stichworten,
- eine Checkliste mit Fragen für den Projektalltag sowie
- eine Liste mit Handlungsalternativen und Denkanstößen für den Ausblick über das Tagesgeschehen hinaus.

Die Zusammenfassungen sind gut gelungen und helfen dem eiligen Leser, das eine oder andere Kapitel zu überspringen. Die Checklisten sind praxisorientiert und können direkt zur Projektsteuerung eingesetzt werden. Das Layout des Buches ist klar und übersichtlich, sodass Checklisten, Praxisbeispiele etc. auch beim Durchblättern leicht erkennbar sind.

Fazit
Sind Sie an der Herstellung von (Standard-) Produkten für einen externen Markt beteiligt und verfügen Sie über Grundkenntnisse der Softwarearchitektur? Dann werden Sie den größtmöglichen Nutzen aus diesem Buch ziehen können. Liegt Ihr Tätigkeitsschwerpunkt jedoch eher in der Erstellung kommerzieller Softwaresysteme für einzelne Unternehmen, so wird Ihnen die erste Hälfte des Buches nur wenige Impulse geben können. Für Sie wird es erst im hinteren Teil des Buches spannend, wenn die Technik stärker im Vordergrund steht als der Absatzmarkt.

Wer eine Einführung in das Thema Softwarearchitektur sucht oder ausschließlich an der Technik interessiert ist, sollte die Finger von „Beyond Software Architecture" lassen. Wollen Sie aber jenseits von Entwurfsmustern und UML-Diagrammen mehr über die *wirtschaftlichen Erfolgsfaktoren* von Softwarearchitekturen wissen, dann wird Ihnen dieses Buch viele neue Denkanstöße bieten. ■

Luke Hohmann, Beyond Software Architecture: Creating and Sustaining Winning Solutions, Addison-Wesley, 2003, ISBN 0-201-77594-8, 304 Seiten, ca.44,90 €