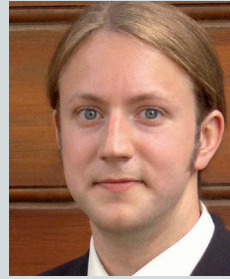




□ Michael Deynet

[E-Mail: michael.deynet@tu-clausthal.de]

ist wissenschaftlicher Mitarbeiter am Lehrstuhl für Software Systems Engineering an der Technischen Universität Clausthal. Er ist dort an der Weiterentwicklung des V-Modell XT, der verbindlich vorgeschriebene Entwicklungsprozess für IT-Projekte der Bundesrepublik Deutschland, beteiligt. Herr Deynet hat in diesem Rahmen sowohl industrielle Projekte als auch Projekte der öffentlichen Hand betreut. Er beschäftigt sich wissenschaftlich u. a. mit modellbasierter Software-Entwicklung, modellbasierten Spezifikations-techniken und der Modellierung von Software-Entwicklungsprozessen.



□ Sebastian Herold

[E-Mail: sebastian.herold@tu-clausthal.de]

ist wissenschaftlicher Mitarbeiter und Doktorand am Lehrstuhl für Software Systems Engineering an der Technischen Universität Clausthal. Seit 2007 beschäftigt er sich dort in Forschung und Projektarbeit mit den Themen Softwarearchitektur und modellbasierte Softwareentwicklung. In seiner Promotion befasst er sich mit der Überprüfung von Sollarchitekturen während Design und Implementierung.

objektspektrum themenspecial: architekturen

Handeln und verstehen - Was für Softwarearchitekten heute zählt!

Die meisten Entwickler werden wohl zustimmen, wenn man sagt, dass jede Zeile selbst geschriebenen Codes und jede Stunde, die mit Lesen (und Verstehen) fremden Codes zugebracht wird, das Verständnis einer Sprache, eines Frameworks oder einer Technologie stärker fördert, als bloßes Bücher wälzen. Das eigene „Hände schmutzig machen“ bleibt eben noch am besten und nachhaltigsten im Gedächtnis hängen, wie schon Konfuzius wusste: „Ich höre und vergesse. Ich sehe und erinnere mich. Ich handle und verstehe.“

In der Ausbildung von Informatikern an Hochschulen geschieht dies von Anfang an durch Übungen und Praktika, wenn es um die Implementierung und Grundlagen des Entwurfs von Softwaresystemen geht. Der Bereich Softwarearchitektur begegnet Studenten dann zum ersten Mal in den Grundlagen der Softwaretechnik. Hier wird es schwierig, Konfuzius' Weisheit weiter zu befolgen. Praktische Anwendung des Gelernten heißt hier schließlich, ein System mit einer gewissen Mindestkomplexität in einem realistischen Umfeld zu entwerfen.

Diese von uns, den Auszubildenden, gewollte, mindestens vorhandene Komplexität betrifft zum einen natürlich das System selbst, die Menge der umzusetzenden funktionalen und erst recht nicht-funktionalen Anforderungen. Aber auch eine realistische Projektgröße und -komplexität sollte gewährleistet sein, beispielsweise

durch das Bereitstellen von Stakeholdern (Kunden, Projektleiter, Betriebe etc.), mit denen der Softwarearchitekt zu tun hat. An der Hochschule ist dies nur zu einem gewissen Grad zu gewährleisten.

Dieser Umstand beinhaltet einige Implikationen für den in der industriellen Praxis arbeitenden Softwarearchitekten. Zum einen wird er seine wertvollsten und umfangreichsten Erfahrungen im praktischen Einsatz über viele Jahre in unterschiedlichsten Projekten machen. Für den gestandenen Softwarearchitekten bedeutet dies aber etwas Weiteres: zu seinen so wieso schon umfangreichen Aufgaben kommt die oft nicht bewusst gemachte Aufgabe des „Weitergebens“ von Wissen und das Coaching angehender Softwarearchitekten hinzu. Schließlich tragen erfahrene Softwarearchitekten, wie alle berufserfahrenen Mitarbeiter, entscheidend zum geistigen Kapital eines Unter-

nehmens bei. Es ist im Interesse des Unternehmens, dass dieses Wissen weitergegeben wird.

Häufig sieht man, dass erfahrene Entwickler irgendwann in den Status des Designers und Softwarearchitekten gehoben werden – sicherlich ein möglicher Weg. Aber es ist entscheidend, dass hier erfahrene Softwarearchitekten die oben genannte Aufgabe gewissenhaft ausführen und durch ihr konzeptionelles Wissen und ihre Erfahrungen an den Stellen unterstützen, an denen ein Neuling entsprechendes Wissen noch nicht besitzt bzw. besitzen kann. Wichtig dabei ist sicher auch, den neuen Softwarearchitekten „einfach mal machen zu lassen“ – siehe Konfuzius!

Das Coaching neuer Architekten kann unter Umständen viel Aufwand bedeuten. Allerdings bekommt der „erfahrene Hase“ auch viele Hilfsmittel durch früher geleistete Arbeit gratis dazu. Gute Architek-

turdokumentation früherer Projekte beispielsweise, die getroffene Architekturentscheidungen, verwendete Architekturprinzipien, -muster und -rahmenbedingungen beschreiben, können eine unschätzbare Quelle für neue Architekten sein. Zusammen mit den Erfahrungen, die sie dann im Laufe ihrer ersten eigen aktiv gestalteten Entwicklungen machen, ist dies ein gutes Fundament.

In vielen Unternehmen geschieht diese „Weiter“-Ausbildung von Softwarearchitekten – wenn auch implizit – durch das Zusammenarbeiten von erfahrenen und weniger erfahrenen Architekten in Teams. Ein explizites Bewusstsein, dass dies eine wichtige Aufgabe von Softwarearchitekten ist, kann diesen Wissenstransfer mit Sicherheit noch effektiver gestalten.

In diesem Zusammenhang ist es hilfreich, wenn zwei Softwarearchitekten mit denselben Konzepten das Gleiche verbinden. In der Softwarearchitektur gibt es aber noch immer unterschiedliche Vorstellungen von Begriffen, Konzepten und Vorgehensweisen. So wendet jeder Architekt in seinen Projekten ein anderes Vorgehen für den Architekturentwurf an. **Dr. Rudolf Hauber** und **Bertil Muth** beschreiben in ihrem Artikel einen Architekturprozess, der es ermöglicht, den Entwurf von Softwarearchitekturen projektübergreifend (und architektenübergreifend) zu vereinheitlichen.

Eine weitere Möglichkeit besteht darin, durch Zertifizierungen zum Softwarearchitekten eine Vereinheitlichung zu erzielen. Ein entsprechendes Zertifikat bescheinigt dem Inhaber gewisse Kenntnisse und einen nachvollziehbaren Wissensstands zum Thema Softwarearchitektur. **Wolfgang Fahl**, **Phillip Ghadir** und **Mahboub Gharbi** berichten in ihrem Artikel vom Sinn und Unsinn solcher Zertifizierungen.

Dr. Darius Silingas widmet sich auf pragmatischer Ebene einem beliebtem Reizthema: wie verträgt sich Architektur bzw. Architekturdokumentation mit agilen Ansätzen wie Scrum? In diesen findet eine Dokumentation der Gesamtarchitektur oftmals nicht statt. Dass dies nicht sein muss, sondern eine entsprechende Dokumentation mit heutigen UML-Modellierungstools zügig erstellt werden kann, wird in seinem Artikel erläutert.

Eine der Hauptfähigkeiten eines Softwarearchitekten ist sicherlich, Architektur angemessen kommunizieren und dokumentieren zu können. Diese Fähigkeit ist insbesondere in einem verteilten Team, im extremsten Fall global verteilt, von großer Bedeutung. **Boris Wehrle** beschreibt in seinem Artikel, wie Werkzeuge wie Visual Studio 2010 helfen können, dass Architekturdokumentation „gelebt“ werden kann.

Hubert Zenner befasst sich in seinem Artikel mit der Nachvollziehbarkeit von sich ändernden Anforderungen in der Architektur und betrachtet dabei insbesondere die Werkzeugunterstützung von IBM Rational.

In Unternehmen besteht der Bedarf, ein System nicht nur isoliert zu entwickeln, sondern auch in eine Systemlandschaft einzubetten. **Wolfgang Keller** greift dies auf und beschreibt das Thema Unternehmensarchitektur und die Rolle des Unternehmensarchitekten.

Die Größe und Funktionalität von Systemlandschaften nimmt stetig zu. Während der Entwicklung eines Systems in einer Systemlandschaft verändern sich das Verhalten oder die Schnittstellen von anderen Systemen. Diese Veränderung ist während der Entwicklung eines Systems nicht planbar. Hierbei stoßen klassische Verfahren für den Architekturentwurf an ihre Grenzen. **Prof. Andreas Rausch**, **Prof. Ursula Goltz**, **Prof. Jörg P. Müller** und **Dirk Niebuhr** geben in ihrem Artikel eine Einführung in diese Problematik. ■

Wir wünschen Ihnen eine abwechslungsreiche und interessante Lektüre dies Online-Themenspecials!

Michael Deynet und Sebastian Herold