



□ Michael Mahlberg

[E-Mail: mm@michaelmahlberg.de]

ist Inhaber und Vorstand der Consulting Guild AG und zurzeit auf weltweiter Wanderschaft nach den Regeln der Walz. Abgesehen davon hilft er Unternehmen seit dem letzten Jahrtausend dabei, Softwareentwicklung und -nutzung zu verbessern. Seit 1985 ununterbrochen selbstständig, hat er viele Technologien kommen und gehen sehen, aber noch kein Jahr gesehen, in dem Architekturen und Entwicklungsprozesse nicht zentral in seiner Tätigkeit gewesen wären.

objektspektrum themenspecial: architekturen

Softwarearchitektur oder die Neuerfindung des Rades?

Obwohl es manchmal so aussehen mag, als seien alle großen Fragen der Architektur von „informationsverarbeitenden Systemen“ schon oft gelöst worden, ist das Thema Softwarearchitekturen heute mindestens so lebendig und wichtig wie vor 60 Jahren. Im Unterschied zu damals stehen uns heute aber ganze Büchereien von Referenzarchitekturen und eben jene 60 Jahre an Erfahrung seit der Auslieferung des ersten kommerziellen Computers zur Verfügung. Wie die Welt des „IT“-Architekten im weiten Feld von eingebetteten Systemen bis zu Unternehmenssoftware in der Cloud heute aussieht, welchen Stellenwert Hardwareüberlegungen heute noch haben und welche aktuellen Strömungen hilfreich sein können, beleuchtet diese Ausgabe des Online-Themenspecials.

Die Forderung, das Rad nicht neu zu erfinden, ist ein Argument, das häufig in die Diskussion geworfen wird. Besonders wenn in neuen Projekten die Frage nach dem Aufwand für Architekturaufgaben aufkommt oder die Architektur bestehender Projekte hinterfragt wird. Worin bestehen denn die aktuellen Herausforderungen für Softwarearchitekten?

Die Restriktionen im Kleinen ...

Haben Softwarearchitekten heute beispielsweise noch mit den gleichen Problemen zu tun, wie diejenigen, die die Systeme für die Mondlandung entworfen haben? Im Enterprise-Umfeld habe ich dazu oft die Aussage gehört – und zugegebenermaßen auch vertreten –, dass die Anforderungen ja heute ganz andere wä-

ren, ganz andere Ressourcen zur Verfügung stünden und selbst manche Armbanduhren heute schon über mehr Speicherkapazität verfügen würden als der Navigationsrechner der Mondfähre mit seinen 4 Kilobyte.

Das mag zwar stimmen, aber wer entwirft die Software für solche Armbanduhren? Und braucht die Software für die Steuergeräte im Auto keine Architektur, nur weil diese Steuergeräte es zwar vielleicht an Rechengeschwindigkeit und Speicherplatz mit den Bordrechnern der ersten Raumschiffe aufnehmen können, aber schon im Vergleich zum allerkleinsten Rechner heute beim Discounter zu erstehenden Rechner um mindestens eine Größenordnung zurück liegen?

Konkrete Informationen für diejenigen, die sich in dieser Welt bewegen, und interessante Einblicke für „Außenstehende“ (wie mich) bieten dazu die Beiträge von **Peter Maurer**, der den Einsatz von Frameworks in eingebetteten Systemen beleuchtet und die übergreifende Diskussion und Bewertung der einzelnen Aspekte von „Softwarearchitektur in eingebetteten Systemen“ im Artikel von **Dr. Michael Sturm** und **Dr. Alexander Harhurin**.

Während unser Komfort und unsere Sicherheit auch von guten Architekturen in eingebetteten Systemen abhängen, ist am anderen Ende des Spektrums, bei Systemen die um die ganze Welt verteilt werden, eine neue Herausforderung für Architekten gewachsen.

... und die Herausforderungen im Großen

Aktuelle Themen, wie die Verlagerung von Anwendungen in „die Cloud“, stellen die Architekten vor neue Herausforderungen, besonders wenn es sich um Legacy-Anwendungen handelt. Dies macht der Artikel von **Karlheinz Peter** und **Dr. Hans-Joachim Pross** deutlich. Das aber auch eine evolutionäre Annäherung an diese Welt gut möglich ist, wenn man passende Muster verwendet hat, und welche Stolpersteine und Höhepunkte auf diesem Weg liegen können, zeigt der Erfahrungsbericht von **Boris Wehrle**.

Vermittler zwischen den Welten

Egal, ob bei Systemen in der weltweiten Wolke, in der Welt der echtzeitsicheren Kleinstsysteme oder irgendwo dazwischen – es ist nicht alleine mit dem isolierten Entwerfen der Architektur getan. Vielmehr ist das Entstehen einer Architektur ein Kommunikationsprozess, der insbesondere in Projekten mit agilem Hintergrund von vielen getragen und permanent gelebt werden muss.

Dr. Horst Kargl zeigt in seinem Beitrag von ihm beobachtete Kommunikationsfallen auf und gibt hierfür Lösungsvorschläge. Um eine Koexistenz auf ganz anderer Ebene kümmert sich **Udo Pracht** in seinem Artikel zu DevOps. Die unterschiedlichen Interessen von Anwendungs- und Infrastrukturarchitekten haben immer wieder zu mehr oder weniger zermürbenden, auf jeden Fall aber „zeitfressenden“ Diskussionen zwischen denen, die im Unternehmen für den Betrieb der Systeme zuständig sind, und denjenigen, die neue Systeme bauen, geführt. Der Bei-

trag zeigt auf, wie es vielleicht in Zukunft anders gehen kann.

Reality Check

Nicht nur an der Verbindungsstelle zwischen Entwicklung und Betrieb, auch im Übergang zwischen Modell und System haben uns die letzten Jahre neue Erfahrungen gebracht. Die Hoffnung, die Architektur „nur noch“ modellieren zu müssen, um danach auf Knopfdruck Anwendungssysteme nach diesem Muster zu erhalten und sicherzustellen, dass „alles“ fehlerfrei der Architektur entspricht, hat in den letzten Jahren den einen oder anderen Dämpfer bekommen. Im Erfahrungsbericht von **Jürgen Krey**, **Dr. Vladimir Rubin** und **Sebastian Rose** werden nicht nur konkrete Vor- und Nachteile modellbasierter Entwicklung aufgezeigt, sondern auch direkt Lösungen vorgestellt, um eine Koexistenz von generierten und manuell erstellten Bestandteilen der Systemwelt sicherzustellen.

Woher kommen jetzt die Architekturen?

Betriebswirtschaftlich am schönsten wäre es natürlich, wenn man die Architekturen gar nicht mehr entwickeln müsste – und der Gedanke, dass es doch eigentlich mittlerweile fertige Architekturen zum Auswählen geben müsste, ist naheliegend und verführerisch. In der Tat sind viele der Architekturmuster und -prinzipien, die wir heute nutzen, gut dokumentiert und alles andere als neu. Aber eben nur die Architekturmuster, nicht die konkreten Architekturen.

Um von dort zur aktuellen Architektur für das jeweils anstehende Projekt zu kommen, ist immer noch viel zu tun. Was tat-

sächlich zu tun ist, um auch bei wirklich großen und komplexen Systemen zu tragfähigen Architekturen zu kommen und bekannte Muster aus dem Umfeld kleiner und mittlerer Systeme sinnvoll zu adaptieren, zeigen **Phillip Ghadir** und **Stefan Tilkov** in ihrem Beitrag.

Wie unterschiedlich reale, nach aktuellen Architekturmustern und -prinzipien entworfene und gebaute Systeme aussehen können, wird abschließend von **Jens-Christian Fischer** dargestellt.

Dunlop und MacPherson

Hätten John Dunlop, der Erfinder des luftgefüllten Reifens, und Earle MacPherson, der Erfinder der nach ihm benannten Radaufhängung, sich von der Aussage „Wir wollen das Rad doch nicht neu erfinden“ bremsen lassen sollen? Meine Meinung ist hier ein klares „Nein, aber“. Wie für die Softwarearchitekten des 21. Jahrhunderts war es in diesem ganz anderen Bereich des Engineerings nötig, die bestehenden Dinge zu kennen, die neuesten Strömungen mitzubekommen und die Dinge auf dem aktuellen Stand der Technik zu kombinieren – aber ohne alles schon Gelernte zu ignorieren.

In diesem Sinne wünsche ich Ihnen viel Vergnügen beim Studium der Anregungen in diesem Online-Special „Architekturen“ und viele neue Ideen und Erkenntnisse. ■

Ihr



Michael Mahlberg