

[Wie "miraculös" ist Objekt-Orientierung?](#)

[System- und Software-Architektur, das Rückgrat jeder IT-Unternehmung](#)

[Wettbewerb](#)

[Software-Engineering ist unser Metier](#)

[Editorial: Software-Entwicklung gestern und heute](#)

[Info Bridge kostenlos abonnieren / Feedback](#)

Wie "miraculös" ist Objektorientierung?



Man konnte vor ein paar Monaten in der Zeitung über den schnellen Untergang eines als "Stern" an der Börse aufgestiegenen Unternehmens lesen. Dank Informationen über "neuartige Technologie" (Objektorientierung, welche seit 10 Jahren Usanz ist) konnten Investoren zu hohen Investitionen bewegt werden. Die Firma hat kommerzielle Software für KMUs geschrieben (was Schweizer Unternehmer schon vor 10 Jahren mit OO gemacht haben) und hat einen grossen Akzent auf den Begriff Objektorientierung gelegt.

Ich wage zu behaupten, obwohl ich mich kaum in den Gedankengang eines Investors eindenken kann, dass bei den Investoren der Glaube entstanden ist, dass die objektorientierte Technologie zum (Börsen-) Absturz geführt hat. Diese "Netzbeschmutzung" einer seit über Jahrzehnten im Einsatz stehenden Technologie ist bedauerenswert. Aber auch der Glaube, dass der Erfolg eines Softwareunternehmens durch Technologie alleine bestimmt würde, ist ein Irrglaube.



Objektorientierte Technologie, wenn beherrscht, ist zwar in heutigen Software-Projekten nicht mehr wegzudenken. Skalierbare Systeme, ja sogar die graphische Oberfläche ihres PCs, oder die paragrafenorientierte Textverarbeitung, der häufig gebrauchte WEB-Browser und vieles mehr, könnte ohne die objektorientierte Technologie nicht in dieser Qualität (und die kann immer verbessert werden) hergestellt werden.

Die Anwender jeder Technologie sind Menschen und alleine die bestimmen den Ausgang eines Projektes. Können Entwickler nicht mit einem klaren Entwicklungsprozess geführt werden, so endet das Projekt wohl in der Schublade.

Aber auch das Erfassen der Benutzer-Anforderungen (heute zum Beispiel mit dem sogenannten Use-Case Verfahren) ist ein wesentlicher Bestandteil des Lebenszyklus eines Software-Projektes.

Erhält der Kunde nicht was er erwartet, dann wird die



Erreicht der Kunde nicht was er erwartet, dann wird die beste Technologie und der beste Entwicklungsprozess nicht für steigende Kurse sorgen. Und die Prozesse führen Leute aus und es sorgen Menschen dafür, dass ein gesetztes Ziel erreicht wird und dass Hindernisse dorthin ausgeräumt werden.

Erfolgreiche Projekte werden am besten in kleinen dedizierten und gut zusammenspielenden Teams zum Ziel gebracht. Damit man die Aufgaben auf mehrere kleine Teams aufteilen kann, müssen Module, welche nahtlos zusammengefügt werden können, bestimmt werden.

Damit dies gemacht werden kann, muss eine Architektur vorhanden sein (siehe dazu Artikel ~~System- und Software-Architektur, das Rückgrat jeder IT-Unternehmung~~).

Aufgrund der Architektur kann dann das Organisatorische abgeleitet werden und der geeignete Entwicklungsprozess, die effektivste Technologie und die dazugehörigen Werkzeuge können ausgewählt werden. Klar steht die Technologie und die Werkzeuge im Vordergrund, welche die Teammitglieder bereits kennen. Doch manchmal muss auch auf neue, dem Team unbekannte Werkzeuge oder Technologieteile umgeschwenkt werden. Hier bewährt sich in der Regel eine konservative Einstellung, zum Vergleich die Regel im konventionellen Bergsteigen (nur eine Hand oder ein Fuss frei, der Rest gesichert) und nicht wie im heutigen Freestyle-Klettern ("maximal zwei Finger am Berg" - Übertreibung). Ist dies nicht gegeben, lohnt es sich beim Einsatz einer oder mehrerer neuen Technologien oder Werkzeuge einen Prototyp für die Technologiedurchgängigkeit zu bauen. Das erspart einem viele Überraschungen. Ausbildung und stetige Qualitätssicherung ist ein Teil eines jeden Entwicklungsprozesses.

Eine von mir gemachte Untersuchung von vier etwa gleich grossen Projekten (>2 Personenjahre) - auf Objektorientierung basierend - auf für Erfolg ausschlaggebende Faktoren ergab einen Hauptfaktor: der Mensch (siehe Bericht ~~OO-Projekte im Vergleich - Erfahrungsbericht~~).

Keine allzu neue Erkenntnis, doch werden zu oft die Schlagworte der Technologie als Verkaufsmittel eingesetzt und es wird versucht, von Menschen so unabhängig wie möglich zu werden. Dies ist der falsche Weg und jeder, der kürzlich an der Börse tief in den Geldbeutel langte, hat gemerkt, dass jede andere Behauptung pure Spekulation ist.

Wir wünschen Ihnen viel Spass beim Lesen. Über ein Echo auf unsere Info Bridge freuen wir uns.

Andres Koch, *Object Engineering GmbH*

System- und Software-Architektur, das Rückgrat jeder IT Unternehmung



Würde man ein Haus bauen, das eine Million kostet, ohne den Zuzug eines Architekten oder Baufachmannes? Wohl kaum.

Baut man in der Praxis ein IT-System, welches Kosten in Millionenhöhe verschlingt, ohne den Beizug eines Software- oder System-Architekten? Die traurige Antwort kennen wir leider auch.

Wie kommen Städte mit den wachsenden Transportproblemen, Energiezufuhr und Umweltbedingungen zurecht, welche zu lange auf eine systematische Zonenplanung, Stadt- und Verkehrsplanung verzichten haben und sich stattdessen zu lange mit politischen Querelen verweilt haben?

Und wie sieht es mit der Analogie der Stadtplanung zur IT-Unternehmens-Architektur aus, verweilt man sich nicht gerade hier in grösseren Firmen mit "politischen" Zankereien?



Das Erstellen einer Architektur ist kein "Jekami", gemeint keine Gruppenaktivität. "Gruppenaktivitäten" tendieren zu zuvielen Kompromissen und verwässerter Systematik. So verwegen es tönt, Einzelpersonen erstellen bessere Architekturen, erreichen zwar keine Perfektion, aber kommen näher zur idealen Lösung. Architektur und Doktrin sind nicht weit voneinander entfernt.

Das Systematische muss aber neben aller Kreativität enthalten sein und Komplexität ist fehl am Platz. Im Gegenteil, einfache, aber durchdachte Konzepte mit System haben eine längere Lebensdauer. Dazu gehört in einer IT-Architektur:

- Modularität
- Schnittstellen
- Kommunikations-Infrastruktur vom Netz bis zur Software (Middleware)
- Normen und Richtlinien

Dass so ein Vorgehen kostenwirksam ist, lässt sich nicht verschweigen (Architektur in der Baubranche ist ja auch nicht vernachlässigbar). Man muss sich hier die Frage stellen, wie es mit den Kosten aussieht, wenn man sich um eine IT-Architektur herummogelt. Es entstehen Kosten in vielen Jahren danach, die zwar immer "nur" in einzelnen Projekten anzufallen scheinen, aber von einer mangelhaften oder fehlenden IT-Architektur herrühren. In jedem Projekt wird "das Rad" oft wieder neu erfunden; gleiche Technologien, Methoden und Werkzeuge werden neu evaluiert und erprobt. Anschlüsse an andere Applikationen werden durch zeitaufwendige und umständliche Art komplexer als nötig. Bestehende Komponenten können nicht benutzt werden, da die Schnittstellen nicht verwendbar sind.

Und wenn das alles so einfach wäre, wieso macht man es dann nicht? Sind Technologie und Methodik so schwierig? Nein, dies ist nicht der Faktor. Der einzige Faktor ist der Mensch und seine Kooperationsbereitschaft. Gerade in der IT-Branche ist es erstaunlich, wie viel Arroganz anzutreffen ist, welche eine Zusammenarbeit mit anderen Fachleuten schwierig macht. Wo man Arroganz antrifft, ist dahinter nur zu oft auch mangelndes Fachwissen anzutreffen (in der Medizin genauso wie in IT und anderen Gebieten). Dadurch dass es menschlich ist, seine Schwächen zu verbergen, gleiten Kooperationsversuche in politisches Gemetzel ab. Es müsste ein Albtraum eines jeden Managers sein, zu sehen, wie viel Geld dabei verloren geht. Wenn man dadurch verursachte fehlende Flexibilität und Reaktionsfähigkeit einer IT-Organisation hinzurechnet, welche neuen Businessanforderungen nicht gerecht werden können, dann würde sogar der Verwaltungsrat keinen ruhigen Schlaf mehr finden.

Eine IT-Architektur wird nicht über Nacht entstehen und sie wird nicht entstehen, solange keine Person dafür verantwortlich zeichnet. Auch wird eine im Elfenbeinturm erstellte IT-Architektur, so wohltonend die zu Papier gebrachten Grundsätze auch sind, nicht in die Praxis umsetzbar sein und von den Projektleitern glattwegs in den Wind geschlagen.

Aber man kann immer etwas tun, wenn die Lösung auch unmöglich scheint. Schrittweise beginnen, das Wichtigste zuerst, und die Toleranz eine Verbesserung auch im Nachhinein machen zu dürfen, führt zum Ziel. Eigentlich ist IT-Architektur wie Qualitätssicherung, wozu sie auch gehören sollte. Sie kommt nicht nur in einer Phase des Projektes zum Zug, sondern muss während dem ganzen Projektzyklus gelebt werden. Dazu gehört viel Kommunikation und Information zu anderen Projekten und umgekehrt.



Staatplanung vom Feinsten: Eainburgs New Town - Foto © Georg Gerster

IT-Architektur ist ein Kommunikationsjob. Ohne Kooperation mit den Projekt-Architekten geht gar nichts, sondern sie wird nach Strich und Faden sabotiert. Entwickler wehren sich typischerweise im ersten Moment gegen jede Art von Richtlinien und sehen erst nach einiger Zeit ein, dass sie ihnen dienen. Oft werden IT-Architekturen auf einer zu abstrakten Ebene angesetzt. Viele Architekten stellen sich vor, dass nur allumfassende

und bis in Detail geplante Architekturen von Interesse sind. Es ist aber für eine erfolgreiche Architektur wichtiger, dass sie von den Ausführenden (Projektmitarbeiter) verstanden wird, also kommuniziert werden kann und dies auch wird. Vom Groben ins

verstanden wird, also kommuniziert werden kann und dies auch wird. Vom Groben ins Detail hilft auch hier.

Dazu gehören:

- "Landkarte", das Applikationslandschaftsbild des Unternehmens
- Eine Kommunikationsplattform mit "Zonenplan" und Ausbaustrategien für die Applikationen (Transportwege im übertragenen Sinne)
- Normen für die nach aussen publizierte Applikations-Schnittstellen
- Datenmodelle für firmenrelevante Daten und in einem weiteren Schritt die Business-Klassenmodelle, welche ersteres miteinschliessen
- Definition von Darstellungsmethoden (z.B. UML, RUP)
- Definition der Basis-Technologien, wie Kommunikationsmiddleware, Programmiersprachen
- Satz von Basiswerkzeugen, wie Datenbank-Systeme, Entwicklerwerkzeuge, Dokumentationswerkzeuge, Middleware, CASE-Tools, Changemanagement Tools, Betriebssysteme, Sprachen. Dies hat aber eher geringere Wichtigkeit.

Auch wenn die gewählte Technologie und die Werkzeuge nicht von höchster Wichtigkeit sind, darf man die Kosten nicht vergessen. Ein falscher Strategieentscheid bei der Technologie, welche nur von einem Hersteller verfügbar ist und nicht mindestens auf de facto-Standards beruht, kann ein IT-Unternehmen zu einem späteren Zeitpunkt in hohe Migrationskosten stürzen. Oder wenn man auf ein Produkt setzt, bei dem die Laufzeit-Lizenzkosten im Laufe der Zeit in schwindelerregende Höhen schießen, kann das Kosten zum explodieren bringen. Was in der Elektronikindustrie klar gefordert ist, sollte auch in der IT-Branche fassen, nämlich ein Second-Source- oder Third-Source-Hersteller eines Produktes zu haben. Dies ist natürlich besser möglich, wenn auf offene Industrie-Standards gesetzt wird, wie die von W3 (XML), OMG (CORBA) u.a. Aber wie sieht dies beispielsweise bei JAVA aus, eine Sprache, die Sun Microsystems ihr eigen nennt? Auch da gibt es gewisse Risiken.

Nun, das sind nur einige Aspekte zu IT-Architektur. Wichtig dabei ist, dass man daran geht und in enger Kommunikation mit der Geschäftsleitung (Zielsetzer für das Unternehmen) und mit den internen und externen Applikationslieferanten einen Plan (Siedlungsplan) und gewisse Normen vereinbart und all dies dem laufenden Bedarf und Bedürfnissen des Unternehmens zweckmässig anpasst.

Andres Koch

Software Engineering ist unser Metier




Sechs gute Gründe für Projekt-Outsourcing

1. Anforderungsanalyse wird für Sie in Ihrem Betrieb gemacht
2. Architektur/Design durch erfahrene Software-Ingenieure
3. Programmierung und Dokumentation durch geübte Entwickler mit effizienten Werkzeugen
4. Einsatz bewährter und moderner Technologien (C++, Java, CORBA, EJB, JMS u.a.)
5. Integration/Installation auf ihrem Zielsystem
6. Professioneller Support auch nach der Installation gewährleistet

Kontaktieren Sie uns noch heute:

Object Engineering GmbH
Birmensdorferstr. 32

Tel  044 400 47 00
Fax 044 400 47 07

8142 Uitikon-Waldegg

[Kontaktformular](#)

Wettbewerb



Wer über die von uns ausgewählte Brücke fährt, kommt zu einem bestimmten Pass in der Schweiz.

Wie heisst dieser Pass?





Über diese Brücke fährt man zum Simplonpass (Brig-Domodossola)

Editorial



Liebe Leserin
Lieber Leser



Wenn ich in die Tasten lange und mit Java ein Programm oder ein Framework schreibe, dann ist dies für mich eine "Sonntagsbeschäftigung", mit anderen Worten eine Freude die Arbeit zu verrichten. Wenn jemand Teil der Informatik-Vergangenheit aus eigener Erfahrung kennt, wird er Sprache und Werkzeuge der heutigen Zeit zu schätzen wissen.

1973 hatte ich den ersten Kontakt mit der Programmiersprache Fortran (Formula Translation) als Elektroingenieur. Dann folgte ein gewisser Rückschritt beim Eintippen der Maschinencodes am Prozessrechner oder am ersten, selbstentwickelten Mikrocomputer. Da spielte sich die Arbeit hauptsächlich im Hexadezimalcode ab und C3 war so bedeutsam wie GOTO in Fortran. Fortran wurde dann verwendet um Cross-Assembler zu schreiben, die Codes für den Mikrocomputer generierten, um später sich selbst (sog. Bootsstrapping) in den eigenen Code zu übersetzen. Dann folgten bessere Debuggers, Betriebssysteme und Editoren, die damit geschrieben und übersetzt wurden. Programmieren in Pascal, C und später C++ war für einen Assembler Programmierer eine Wohltat von unschätzbarem Wert.

Zwischen früherer Programmierung und der heutigen, welche durch komfortable Tools unterstützt wird, liegen Welten und eine Zeitspanne von 25 Jahren. Einfache Tools sind oft handlicher als die komplexen Supertools. Was aber auch heute noch geblieben und in der Zukunft weiter bleibt, ist die nötige Disziplin qualitativ gute, verständliche und robuste Programme ohne unnötige Komplexität zu schreiben. Aber eben, mit Java macht es bisher am meisten Spass!

Mit freundlichen Grüßen

Andres Koch
Dipl. El. Ing. HTL, M. Math
Geschäftsführer
Okject Engineering GmbH