

## „Komplexe und hochwertige Business-Software einfach, schnell und preiswert hergestellt“

„Our enemy is complexity, and it's our goal to kill it!“

Jan Baan  
Gründer und Ex-CEO BaaN  
1999

**qinet GmbH**  
Mombacher Weg 6 b  
D-65936 Frankfurt am Main

[www.qinet.de](http://www.qinet.de)

---

## Inhalt

<b>1</b>	<b>Zusammenfassung</b>	<b>3</b>
<b>2</b>	<b>Status Quo in der Software-Entwicklung</b>	<b>3</b>
2.1	Komplexität der Softwareentwicklung / Konventionelle Ansätze	3
2.2	Moderne Ansätze zur Lösung	4
<b>3</b>	<b>Der qinet Ansatz</b>	<b>5</b>
<b>4</b>	<b>Die Mission der qinet</b>	<b>6</b>
<b>5</b>	<b>Die Attraktivität von qinet für Partner</b>	<b>7</b>

## 1 Zusammenfassung

Hochwertige Business-Software zeichnet sich dadurch aus, dass sie individuelle Geschäftsprozesse optimal unterstützt, Transparenz schafft und für den Anwender Komplexität reduziert.

Gute Software-Entwicklungsprojekte bedienen sich moderner Methoden und Technologien, schaffen schnell greifbare Ergebnisse und sind flexibel anpassbar, wenn sich Rahmenbedingungen und/oder Anforderungen ändern.

Effiziente Software-Entwicklungsprozesse vermeiden jegliche „Eigen-Komplexität“ und ermöglichen durch den konsequenten und pragmatischen Einsatz von bewährten und weiterentwickelten Techniken und Vorgehen die Herstellung individueller und preiswerter Lösungen.

Die qinet nutzt konsequent innovative, durchgängige Methoden und Werkzeuge, die sich von altbewährten Vorgehensweisen deutlich abheben und empfiehlt sich als verlässlicher und kompetenter Partner für die Konzeption und Entwicklung individueller Business-Software.

## 2 Status Quo in der Software-Entwicklung

### 2.1 Komplexität der Softwareentwicklung / Konventionelle Ansätze

Die Komplexität von Softwareprojekten ist in den vergangenen Jahren sprunghaft gestiegen. Eine Entwicklung die sich auch zukünftig fortsetzen wird, wenn man den aktuellen Trends in der Software-Entwicklung Glauben schenken darf.

Als ein Beispiel auf der technischen Seite sei hier der zurzeit in aller Munde befindliche SOA Architekturansatz genannt, dessen durchgängiges Verständnis im engeren Sinne aber nur noch wenigen Experten zugänglich ist.

Signifikant für diesen Ansatz ist eine Grundidee, die sich auch in anderen technologischen Konzepten wiederfindet:

Sie begegnen einer steigenden fachlichen Anforderungskomplexität mit nicht minder komplizierten technischen Standards und versuchen darüber hinaus, komplexe Business-Prozesse mit ebenso vielschichtigen Projektstrukturen beherrschbar zu machen.

In der Hoffnung der Komplexität und dem stetig zunehmenden Zeit- und Kostendruck Herr zu werden, kommen große Projektteams zum Einsatz. Als Folge steigt der Kommunikationsaufwand zur Abstimmung, sowohl projektintern, als auch gegenüber den zuständigen Fachbereichen und häufig sind die Fähigkeiten aller Teammitglieder im Schnitt nicht ausreichend, um der wachsenden Komplexität effektiv entgegen zu wirken.

Als Vorgehensmodell kommt bei Software-Projekten zumeist das etablierte Wasserfallmodell zum Einsatz, das sich durch eine einfache Struktur und fest definierte Phasen auszeichnet. Diese festgelegte Struktur wird dem Modell mit steigender Komplexität der Projekte mehr und mehr zum Verhängnis und ist für viele fehlgeleitete Projekte verantwortlich. Die Übergänge zwischen den einzelnen Phasen bilden scheinbar unüberwindbare Hindernisse für den

Informationsfluss. So kommt es, dass sich Fachberater während der Konzeptionsphase mit technischen Restriktionen beschäftigen, während die Entwickler während der Software-Entwicklung die genauen Details der Unternehmensprozesse kennen müssen.

Dazu kommt, dass die Fachabteilung zur Beschreibung ihrer Anforderungen andere Methoden und Werkzeuge als die Entwicklungsabteilung benutzt, was in Form eines Medienbruchs die Kommunikation weiter erschwert. Während die Fachabteilung ihre Anforderungen vorwiegend auf der Basis Ihrer Geschäftsprozesse beschreibt, liegt der Schwerpunkt in den Entwicklungsabteilungen überwiegend in der Analyse von Objekt- und Datenstrukturen. Hieraus resultieren Abstimmungs-Konflikte und der Abgleich von Inhalten und unterschiedlichen Informationsständen führt zu Mehraufwand.

Im Ergebnis erweisen sich komplexe Softwareprojekte sehr oft als nicht mehr beherrschbar. Sie scheitern und werden teilweise in einem Projektstadium komplett verworfen in dem bereits immense Ressourcen aufgewendet wurden.

## 2.2 Moderne Ansätze zur Lösung

Diesen Nachteilen der herkömmlichen Softwareentwicklung ist zwar in den letzten Jahren mit dem Einsatz innovativer, iterativer Vorgehensmodelle (z.B. RUP oder OOP) und Projektmethodiken (z.B. MDA/MDS und DSM/DSL) begegnet worden, zu einer grundlegenden Vereinfachung insbesondere aus Sicht der Anwender und Fachbereiche ist es dadurch jedoch immer noch nicht gekommen, im Gegenteil.

Durch den Einsatz iterativer Ansätze wird das Grundübel zwar erkannt, deren Einbettung in starre Strukturen und fixierte Abläufe „hebelt“ den Nutzen wieder aus.

Seitens der neueren Implementierungsmethodiken sind als interessanteste die „Model Driven Architecture (MDA)“ und „Domain Specific Modelling (DSM)“ zu nennen.

Mit MDA werden Softwaresysteme unabhängig von der Architektur spezifiziert und mittels UML (Unified Modelling Language) modelliert. Daraus wird in **mehreren** Schritten gegen eine spezifische Architektur / Sprache eine Lösung generiert.

DSM hingegen beruht auf der Idee, eine Sprache (in der Regel eine Untermenge der UML) zu definieren, die speziell für **ein** Aufgabengebiet geeignet ist, um mit dieser die Lösung zu beschreiben. Mit dieser Spezifikation wird mit Hilfe des domänenspezifischen Generators eine Lösung gegen eine Architektur generiert.

Beiden methodischen Ansätzen ist gemeinsam, dass große Teile der Softwareproduktion gegen die Anwendungs-Architektur generiert werden können. Bisher ist es gelungen, mit diesem Ansatz bis zu 40-60% des gesamten Codierungsaufwandes direkt aus Modellen zu gewinnen. Als Schwierigkeit erweist sich eine durchgängige Generierung über alle Architekturschichten (z.B. in einer 3-Tier-Architektur im Bereich Frontend UND Middleware UND Persistenzschicht). Beides zusammen genommen ist der Grund, warum es in letzter Zeit wieder etwas ruhiger um diese beiden viel versprechenden methodischen Ansätze geworden ist.

***Gesucht wird daher immer noch eine Methodik innerhalb eines pragmatischen Projektvorgehens, die einen noch wesentlich größeren Anteil an generierten Code über alle Schichten hinweg ermöglicht, zusammen mit einem den Projekterfolg garantierenden Vorgehen!***

### 3 Der qinet Ansatz

Die Herausforderung für Software-Projekte ist somit, den Prozentsatz an durchgängiger Code-Generierung zu erhöhen und hierbei ein modernes Vorgehen zu etablieren, was beherrschbar bleibt.

Beides ist der qinet zusammen mit ihren Partnern gelungen.

Die qinet fokussiert sich auf die Konzeption und Umsetzung individueller und integrierter IT-Lösungen in hoher Qualität zu Festpreisen. Sie benutzt hierbei die gleiche Architektur, und zwar von hochskalierbaren Enterprise Anwendungen bis hin zu kleinen Installationen.

Dies gelingt durch den konsequenten Einsatz eines in der Praxis bewährten Business Frameworks in Kombination mit einem durchgängigen Generatoransatz.

Hierdurch ist die qinet in der Lage, bis zu 90 % des Codes einer Business-Anwendung über alle Schichten durchgängig zu generieren.

Auch gelingt es durch die durchgängige Generierung über alle Architekturschichten hinweg, dass überhaupt erst ein iteratives Vorgehen - das wir innerhalb eines modernen IT-Projektes als den richtigen Ansatz betrachten - pragmatisch zum Einsatz kommen kann.

Damit wird garantiert, dass zu einem sehr frühen Projektzeitpunkt eine lauffähige Basisanwendung implementiert wird, die zur weiteren fachlichen Abstimmung benutzt wird. Hierbei werden nach 20% der Projektlaufzeit 80% und mehr der Funktionalität in einer inkrementell wachsenden Anwendung frühzeitig zur weiteren Abstimmung bereitgestellt.

Ein wesentlicher Vorteil hiervon ist, dass sich die Betroffenen seitens des Auftraggebers nicht mit unnötig viel Papier herumschlagen müssen. Die Funktionalität wird an der Basisanwendung überprüft. In Abhängigkeit von der Projektgröße finden weitere Abstimmungsrunden statt, bis hin zur Fertigstellung und Produktionseinführung der Anwendung.

Durch die Benutzung des fertigen Business-Frameworks erfolgt eine strikte Trennung der Technik von der Fachlichkeit. Als Folge hiervon können sich die Entwickler komplett auf die Implementierung der Fachlichkeit beschränken, die technische Umsetzung hingegen ist komplett durch das Business-Framework schon erledigt.

Ein weiteres Unterscheidungsmerkmal ist, dass die qinet nur mit kleinen Projektteams arbeitet, die ausschließlich aus Experten bestehen, die seit vielen Jahren erfolgreich zusammen arbeiten und perfekt aufeinander abgestimmt sind.

Die Zusammensetzung der Experten Teams erfolgt über Kooperationspartner mit der qinet als Generalunternehmer. Alle Kooperationspartner sind hoch qualifizierte Spezialisten und verfügen auf Ihrem jeweiligen Fachgebiet über langjähriges Know-How.

In letzter Konsequenz heißt dies, dass die qinet 3 bis 5 mal schneller als der Markt ist, bei gleichzeitig hoher Qualität.

## 4 Die Mission der qinet

Letztendlich beruhen die o.g. Verbesserungen des von der qinet eingeschlagenen Weges auf die konsequente Umsetzung und Beherzigung folgender Prinzipien und Leitsätze:

- Weniger ist mehr
- Einfacher ist besser
- Das Leichte schlägt das Komplizierte
- Das Schnelle ist unschlagbar
- Die Qualität überzeugt
- Der Preis ist konkurrenzlos

Die qinet beschreitet durch die konsequente Einhaltung obiger Prinzipien und Leitsätze andere Wege in der Softwareentwicklung. Gemeinsam mit ihren Kunden gelangt man somit zu einer preiswerten und schnellen Herstellung von komplexer Business-Software.

## 5 Die Attraktivität von qinet für Partner

### **Gesicherter Projekterfolg**

Die Kombination aus Business Framework, automatisierter Codeerstellung und offener Zielarchitektur stellt in Verbindung mit einem iterativen Vorgehen den Projekterfolg von Anfang an sicher.

### **Erhöhte Akzeptanz**

Das schnelle Erstellen der Basisanwendung erleichtert die Kommunikation mit der Fachabteilung. Durch eine frühzeitige Befragung an der Basisanwendung wird die Akzeptanz erhöht.

### **Beschleunigung der Entwicklungszeiten**

Durch den Generierungsansatz in Kombination mit dem zugrunde liegenden Business Framework lassen sich die Projekt-Entwicklungszeiten deutlich verkürzen. Ein schnelleres Time-to-Market ist sichergestellt.

### **Skalierbare Architektur**

Die Zielplattform basiert auf offenen Standards und kann schnell und kostengünstig an neue Anforderungen angepasst werden. Darüber hinaus lässt sich die Anwendung komplett in die Unternehmens-IT integrieren oder gar als Integrationsplattform nutzen.

### **Qualität**

Automatisch generierter Code zeichnet sich im Vergleich zu manuell erstelltem Code durch höhere Qualität aus. Entwickler haben nur wenige Möglichkeiten fehlerhaften Code zu erzeugen

### **Geringe Projektkosten**

Die qinet realisiert Softwareprojekte über 30 Prozent günstiger – bei gleichzeitig höherer Qualität.

### **Reduzierung der Wartungskosten**

Die erhöhte Qualität und die einfache Weiterentwicklung sorgen für eine signifikante Kostenersparnis bei Wartung und Support.