



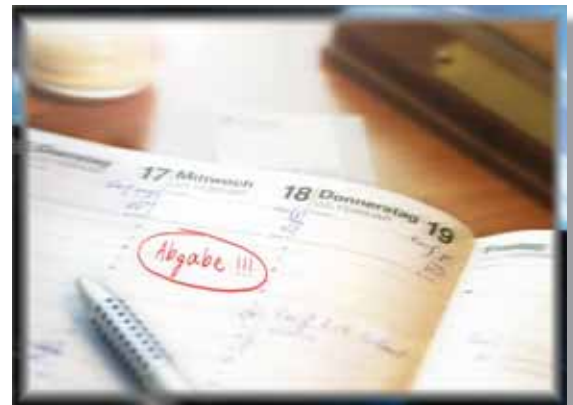
## Software und Zeit

In kaum einem Gebiet weichen Theorie und Praxis so voneinander ab wie im Bereich der Softwareentwicklung. Während die Theorie dazu rät, langfristig zu denken, größere Betrachtungszeiträume zu wählen und die Zukunft einer Lösung bereits am Beginn einer Entwicklung mit einzubeziehen, wird die Praxis von ganz anderen Wünschen bestimmt: Rasche Fertigstellung, billige Lösungen und möglichst frühzeitiger Einsatz erscheinen oft ideal.

Der Unterschied von Theorie und Praxis ist also ganz offensichtlich durch den Zeitrahmen bestimmt, innerhalb dessen eine Softwarelösung betrachtet wird. Die kurzen Managementzyklen, wirtschaftliche Notwendigkeiten und der Mitbewerbsdruck zwingen dazu, kurzfristig nützliche Ergebnisse zu erzielen. Die Folge ist die besagte Ausrichtung in Form einer kurzfristigen Betrachtungsweise, die in der Praxis so oft zu finden ist. Die Gründe dafür bleiben hier unbestritten – schließlich ist es nun einmal eine Realität, dass Unternehmen auch kurzfristig erfolgreich sein müssen. Was bei einer ausschließlich kurzfristigen Betrachtungsweise allerdings unter den Tisch fällt, sind massive Mehrkosten, die sich zu einem späteren Zeitpunkt einstellen. Man denke dabei etwa an das klassische Beispiel der Entwicklungsdokumentation. Ein Softwareprojekt lässt sich ohne Dokumentation kurzfristig wesentlich günstiger durchführen, als wenn begleitend eine gründliche Dokumentation erstellt werden würde. Sollen aber ein, zwei Jahre später Erweiterungen an der Lösung vorgenommen werden, macht sich das Fehlen einer Dokumentation empfindlich bemerkbar.

Die Programmierer sind vielleicht nicht mehr verfügbar oder können sich nicht mehr im Detail erinnern, eine intensive Analyse des bestehenden Codes wird notwendig. Im schlimmsten Fall treibt das Einarbeiten in die bestehende Software die Kosten so in die Höhe, dass eine Erweiterung praktisch einer Neuentwicklung gleichkommt. Mit einer primär kurzfristigen Betrachtung eines Entwicklungsprojekts werden also in die Lösung „Zeitbomben“ eingebaut, die in der Zukunft detonieren und immensen wirtschaftlichen Schaden verursachen. Wesentlich günstiger ist es also, neben der kurzfristigen Rentabilität auch die langfristige Wirtschaftlichkeit zu berücksichtigen. Im Folgenden rufen wir ein paar Grundlagen in Erinnerung, die Ansätze für den richtigen Umgang mit Zeit und Zeitskalen in der Softwareentwicklung geben:

photography: www.gerryfrank.at



*Für langfristig erfolgreiche Softwareprojekte sind vorausschauende Maßnahmen entscheidend.*

**Sichern Sie Ihr Projekt mit einem Pflichtenheft ab.** Ein Pflichtenheft ist nicht nur wichtig, um eine professionelle Beauftragung der internen oder externen Programmierer zu ermöglichen. Auch als Vereinbarung mit den Nutznießern der geplanten Software ist es sehr

nützlich – denn Vorstellungen neigen dazu, sich mit der Zeit schleichend zu verändern. Das gilt ganz besonders für nicht genau spezifizierte Vorstellungen, die von einem größeren Personenkreis entwickelt werden.

### **Implementieren Sie in Richtung Haltbarkeit.**

In vielen Fällen empfiehlt es sich sehr, konsequent in Richtung Haltbarkeit der zu erstellenden Software zu arbeiten. Das kann zum Beispiel so aussehen, dass die Zukunft bewusst mit eingeplant wird und der Code mit dem richtigen Maß an Portierbarkeit ausgestattet wird. Mehr zu diesem Thema finden Sie unter „Portierbarer Code“ auf [www.infrasoft.at/service](http://www.infrasoft.at/service).

### **Führen Sie eine Entwicklungsdokumentation.**

Die Entwicklungsdokumentation ist oft das erste Opfer von kurzfristigen Einsparungen. Wie oben bereits erwähnt, kann sich das Fehlen einer fundierten Dokumentation aber zu einem späteren Zeitpunkt als fatal herausstellen. Die Mindestanforderungen an ein gut dokumentiertes Softwareprojekt finden Sie unter „Dokumentation in der Softwareentwicklung“ auf [www.infrasoft.at/service](http://www.infrasoft.at/service).

**Testen Sie Ihre Ergebnisse.** Auch Tests werden gerne zugunsten einer raschen Fertigstellung vernachlässigt. Das ist fast eine Garantie für einen Misserfolg, da de facto die User zu den Testern werden. Durch die laufende Beschäftigung mit Softwarefehlern entwickeln Sie Vorbehalte, die einer Lösung besonders in der Einführungsphase sehr schaden. Mehr dazu finden Sie unter „Tests in der Softwareentwicklung“ auf [www.infrasoft.at/service](http://www.infrasoft.at/service).

## **InfraSoft als Partner**

bietet Ihnen professionelle Unterstützung.

- **Wir setzen modernste Methoden ein**  
Unsere Mitarbeiter verfügen über fundierte Kenntnisse in C++, Java, SQL, Notes und HTML/XML.
- **Wir stellen Know-how bereit**  
Die InfraSoft-Teams haben langjährige Erfahrungen mit Datenbanken, Networking und Internet.
- **Wir schonen Ihr Budget**  
Unsere Entwickler stehen *standby*. Sie bezahlen uns nur dann, wenn Sie uns brauchen.
- **Wir übernehmen Verantwortung**  
Unsere Teams sind bereit, komplexe Probleme in Eigenregie zu lösen.

**Planen Sie die Wartung im Betrieb.** Für ein langfristig erfolgreiches Projekt ist neben einer einwandfreien Realisierung auch eine Absicherung des laufenden Betriebs entscheidend. Mit vorausschauender Planung können Sie sicherstellen, dass eine Softwarelösung durch Garantie bzw. Wartungsvereinbarungen auch später rundherum gut abgesichert ist.

Wenn Sie mehr über die langfristige Erfolgssicherung von Entwicklungsprojekten erfahren möchten, stehen Ihnen die Profis der InfraSoft unter [info@infrasoft.at](mailto:info@infrasoft.at) gerne zur Verfügung.

**Die Profis von InfraSoft sind interessant für mich. Bitte kontaktieren Sie mich:**

Vorname : \_\_\_\_\_  
Nachname : \_\_\_\_\_  
Firma : \_\_\_\_\_  
Adresse : \_\_\_\_\_

Tel : \_\_\_\_\_  
Handy : \_\_\_\_\_  
Fax : \_\_\_\_\_  
e-mail : \_\_\_\_\_  
via http:// \_\_\_\_\_

