

Keine Angst vor Softwareprojekten!

„Wer wagt, gewinnt – außer er verliert.“

Einige gar nicht so riskante Gedanken von Peter Lieber.

Peter
Lieber



DEFINITIONEN GEFÄLLIG? Ein Risiko ist die kalkulierte Prognose eines möglichen Schadens bzw. Verlustes im negativen Fall (Gefahr) bzw. die Wahrscheinlichkeit des Eintretens eines negativen Ereignisses (mathematisch), eventuell multipliziert mit dem finanziellen Ausmaß (BWL).

Man kann Risiken aber auch aus dem Weg gehen. Hierfür existieren anspruchsvolle Konzepte, und im Gegensatz zu „Unsicherheit“ ist das Eintreten eines Risikofalls zumindest theoretisch kalkulierbar. Fehleinschätzungen von Risiken können jedoch immer auch zu sehr negativen Folgen und sogar zu Katastrophen führen – auch und gerade im IT-Bereich!

Ein bisschen Philosophie?

Dem englischen Mathematiker und Philosophen John G. Bennett zufolge macht erst die Möglichkeit des Versagens die Dinge „wirklich“. So ist echte Freiheit nach

Bennett nur in nicht determinierten Lebenssituationen denkbar, deren möglicher Ausgang also tatsächlich offen ist.

In seinem Hauptwerk „The Dramatic Universe“ behauptet Bennett, dass Momente des „Hasards“ ganz konkret auf physikalischer Ebene zur Grundbedingung des Universums gehören (was auch interessante Querverweise zu Erkenntnissen der modernen Quantenphysik eröffnet). Es gibt keinen perfekten Zustand, der von diesem Zusammenspiel von Unsicherheit und Wille (bzw. Naturgesetzen) frei wäre, da Hasard eine „Grundkonstante“ der Wirklichkeit darstellt.

Konkreter, bitte!

Bei der Softwareentwicklung bzw. bei der Implementierung von Informationssystemen werden zunehmend Methoden des Risikomanagements eingesetzt, um der Komplexität und der damit verbundenen Fehleranfälligkeit zu begegnen.

Aspekte des Risikomanagements sollten über den gesamten Systemlebenszyklus, also beginnend mit dem Konzept über die Entwicklung oder Programmierung, Implementierung und Konfiguration und während des Betriebs bis hin zur Stilllegung des Systems berücksichtigt werden.

Grundsätzlich beginnt Risikomanagement in dem Moment, in dem eine (Software-)Vision, ein Wunschbild der zukünftigen Realität, entsteht. Denn die Chancen, die man dazu wahrnehmen muss, werden durch Unwägbarkeiten gefährdet. Selbstverständlich lässt sich nicht jedes Schadensausmaß in Geld ausdrücken, letztendlich ist mangels einheitlicher Definitionen für „Schaden“ die Bewertung oft subjektiv. Dazu möchte ich gern eine These von Rolf Götz (Deutsche Post AG) heranziehen: „Risikoeinschätzern fällt es schwer, Konsistenz in der eigenen Einschätzung zu wahren – sogar wenn gar keine neuen Informationen zu den Risiken vorliegen.“

Schließlich fällt es uns allen nicht ganz leicht, Zeitverläufe zu bedenken, und zwar in zwei Richtungen:

Kontakt

Die LieberLieber Software GmbH

zählt wie z. B. SparxSystems Software GmbH (Central Europe) zur Firmengruppe von Peter Lieber und agiert als Microsoft Gold Partner mit Windows- und Webkompetenz u. a. in den Bereichen Web, Sharepoint, Intranet, Windows, Office-Erweiterungen sowie Robotersteuerung. Bemerkenswert sind weiters die speziellen Produkte für Menschen mit besonderen Bedürfnissen (z. B. im Fall von Sprachstörungen), Scheduled Mediaplayer für Mediawalls, elektronische Bilderrahmen u. v. a. m.

Treffen Sie LieberLieber

beim Microsoft Big>Day am 31. 3. in Wien!
www.lieberlieber.com
www.sparxsystems.at
peter.lieber@lieberlieber.com

a) Wir tendieren dazu, uns NICHT anzusehen, wie sich eine Größe mit der Zeit verändert, und ziehen daraus also auch keine Schlüsse.

b) Ändert sich eine Größe NICHT, tendieren wir dazu, sie dennoch für variabel zu halten.

Es findet immer ein Abwägen der notwendigen Aufwände zur Risikominimierung oder des (mit der Eintrittswahrscheinlichkeit gewichteten) Schadens bei Risikoeintritt statt. Teilweise werden im Risikomanagement auch Themen des sogenannten „Issue Management“ – das ist die Behandlung eingetretener und vorher nicht identifizierter Risiken – behandelt.

Aus dem Alltag (nicht nur) bei LieberLieber erinnere ich mich an oft langwierige Diskussionen: Risiken, die letzte Woche noch „abschließend“ diskutiert zu sein schienen, werden erneut lang und breit besprochen. Das scheint mir allerdings in der Natur der Sache zu liegen, weil sich ja auch „im realen Leben“ neue Informationen zu den Risiken ergeben. <<

Die Toprisiken

in der Softwareentwicklung ...

1. zu wenig Leute
2. unrealistische Kosten- und Terminpläne
3. falsche Funktionalität
4. falsche Benutzerschnittstelle
5. Vergoldung (überflüssiger Luxus)
6. sich ständig verändernde Anforderungen
7. Probleme mit zugekauften Komponenten
8. Probleme mit extern vergebenen Aufträgen
9. Nichterreichen der verlangten Leistungen (z. B. Reaktionszeit)
10. Überforderung der Mitarbeiter in Bezug auf ihr softwaretechnisches Können

... und was man dagegen tun kann:

1. mehr Leute
2. realistische Kosten- und Terminpläne!
3. richtige Funktionalität
4. richtige Benutzerschnittstelle
5. Es muss funktionieren!
6. „Was genau will ich bekommen?“
7. bis 10. mehr Know-how
(am besten bei bzw. von LieberLieber)