

## Risikosystematik

Als Grundlage für das Risikomanagement in der Softwareentwicklung wurde zunächst eine strukturierte Risikosystematik im Rahmen der Risikoidentifikation und -bewertung erarbeitet (siehe Abbildung 1). Sie wurde in Analogie zur fundierten Darstellung von Göcke (GÖCKE 2002) für das Bauwesen entworfen.

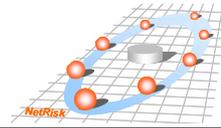


Abbildung 1: Darstellung Risikosystematik

Dabei wird das Projektrisiko heruntergebrochen auf das Ergebnisrisiko und das Vergütungsrisiko. In das Ergebnisrisiko fließen die Faktoren (im folgenden Hauptprojektelemente) Leistungssoll, Qualität, Projektzeit, Beteiligte, IT-Strukturen, sowie Gesetze und Vorschriften ein. Auf das Risiko, welches die Vergütung für die erbrachte Leistung beeinflusst, beziehen sich die Hauptprojektelemente Vergütungssoll, Preisermittlung und Zahlung. Diese Hauptprojektelemente werden im Folgenden näher beschrieben und unter Betrachtung verschiedener zugehöriger Risikogruppen weiter verfeinert.

### Hauptprojektelement: Leistungssoll

Der Auftragnehmer schuldet dem Auftraggeber die vertraglich festgelegte Leistung. Er trägt die Verantwortung für den Erfolg des Projektes. Dabei bestimmt der Vertrag den Umfang und Inhalt der Leistung, die der Auftragnehmer zu erbringen hat, und somit das Leistungssoll. Da sich Software-Entwicklungsprozesse gerade dadurch auszeichnen, dass viele Leistungsände-



rungen unterschiedlicher Form auftreten, bedeutet ein im Vertrag festgeschriebenes Leistungssoll häufig nicht, dass dieses im Verlauf der Projektabwicklung unverändert bleibt. Dies führt für den Auftragnehmer zu zwei grundsätzlichen Problembereichen. Auf der einen Seite muss die Leistung bei Vertragsschluss möglichst klar und eindeutig beschrieben sein, auf der anderen Seite ist darauf zu achten, dass Leistungsänderungen für den Auftragnehmer keine nachteiligen Folgen haben.

Im Folgenden werden die Risiken dargelegt, die sich aus diesen beiden Punkten - Festlegung des Leistungssolls bei Vertragsschluss und Änderungen des Leistungssolls im Projektverlauf - ergeben.

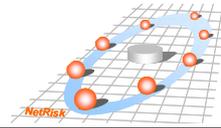
### **Projektelement: Leistungssoll bei Vertragsabschluss**

#### **RiGr1: Unklares Leistungssoll**

Das Leistungssoll spiegelt die Gesamtheit der Vertragsbestandteile wieder. Die interessante Fragestellung in diesem Kontext ist die des Tragens des Risikos falls die Leistung unklar, lückenhaft, falsch oder widersprüchlich beschrieben ist. Es stellt sich die Frage, welche Vertragspartei die daraus entstehenden Risiken zu tragen hat. Unklarheiten über das Leistungssoll können sich aus Widersprüchen in den Vertragsunterlagen oder aus lückenhaften, missverständlichen oder irreführenden Leistungsbeschreibungen ergeben.

#### **RiGr2: Produkthaftung**

Unter Produkthaftung versteht man die Haftung auf Schadenersatz für die Lieferung einer fehlerhaften Kaufsache und für Schäden, die dadurch an anderen Rechtsgütern entstehen. Das Produkthaftungsgesetz beruht auf der EG-Richtlinie 85/374 EG. Ein fehlerhaftes Produkt wird darin definiert als eine hergestellte Sache, die fehlerhaft ist und gewerblich in den Vertrieb gelangte. Zwar gilt dieses zurzeit noch nicht für Software, allerdings wird immer wieder ein angelehntes Haftungsgesetz gefordert (wie z. B. vom Sicherheitsexperten Bruce Schnei-



er<sup>1</sup>). Somit scheint diese Thematik in Zukunft zu einem Risiko für die SW-Industrie zu werden.

### RiGr3: Komplettierungsverpflichtung

Falls anstelle einer umfassenden Leistungsbeschreibung oder über die Leistungsbeschreibung hinaus ein Leistungsziel den Leistungsinhalt bestimmt, besteht eine Komplettierungsverpflichtung des Auftragnehmers. In diesem Fall schuldet der Auftragnehmer alle Leistungen, die für die Erreichung des Leistungsziels notwendig sind.

### **Projektelement: Änderungen nach Vertragsabschluss**

#### RiGr4: Geänderte und zusätzliche Leistungen

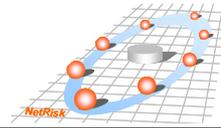
Während des Projektes kann es von Seiten des Auftraggebers zu Änderungen des Leistungsinhaltes kommen. Dabei können bereits bestehende Komponenten des vereinbarten Leistungskataloges geändert oder neue Bestandteile hinzugefügt werden. Beide Arten können zu schwerwiegenden Komplikationen im Projektablauf und im Projektergebnis führen. Dabei gilt die Faustregel: Desto später die Änderungswünsche bekannt werden, desto größer der Änderungsaufwand und somit das Risiko für das Projekt. Gerade in einem SW-Entwicklungsprozess wird eine Änderung des Leistungssolls während des Projektes als Hauptrisiko für einen Misserfolg des Projektes angesehen (vgl. STANDISH GROUP 1995).

#### RiGr5: Vom Auftraggeber gekündigte Leistungen

Eine vom Auftraggeber ausgesprochene Kündigung kann sowohl die gesamte Leistung als auch Teilleistungen betreffen. Dabei kann es für den Auftragnehmer z. B. durch Verpflichtungen gegenüber Dritten, leicht zu schwerwiegenden Folgen kommen, die von Seiten des Auftragnehmers im Vorhinein selten einzusehen und abzuschätzen sind.

---

<sup>1</sup> Bruce Schneier ist Gründer und technischer Leiter von Counterpane Internet Security, Inc. Sein neuestes Buch heißt "Beyond Fear: Thinking Sensibly About Security in an Uncertain World". Kontakt: <http://www.schneier.com/>



## RiGr6: Technologieveränderungen

Da die SW-Branche eine hoch dynamische von Innovation geprägte Industrie ist und es immer wieder zu kleinen „technischen Revolutionen“ kommt, muss das zu entwickelnde Produkt in Bezug auf solche Situationen stabil und robust sein. Dabei ist darauf zu achten, dass neue Entwicklungen rechtzeitig erkannt und das Produkt gegebenenfalls um z. B. entsprechende Schnittstellen erweitert wird.

### **Hauptprojektelement: Qualität**

Qualität ist in DIN EN ISO 9000 (vgl. DIN 2005) definiert als Grad, in dem ein Satz inhärenter Merkmale Anforderungen erfüllt. Dabei werden Anforderungen als Erfordernisse oder Erwartungen angesehen, die festgelegt, üblicherweise vorausgesetzt oder verpflichtend sind. Dabei sind festgelegte Erfordernisse vertraglich oder gesetzlich bereits spezifizierte Anforderungen. Dahingegen müssen vorausgesetzte Erfordernisse zunächst festgestellt und festgelegt werden.

### **Projektelement: Mangel-/ Fehlerfreiheit**

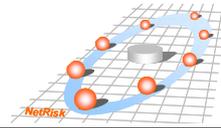
#### RiGr7: Mangel/ Fehler

Nach dem Text der Qualitätsmanagement-Norm ISO 9000 liegt ein Mangel immer dann vor, wenn ein Arbeitsergebnis zu seinem vorgesehenen Zweck nicht in vollem Umfang geeignet ist. Dabei ist es unerheblich, ob die fehlenden Leistungsmerkmale in Form einer Anforderung dokumentiert worden sind oder nicht. Sollten Anforderungen gar nicht erfüllt worden sein, so spricht die Norm auch von Fehlern.

### **Projektelement: Ausführungsqualität**

#### RiGr8: SW-Entwicklungsverfahren

Die Qualität des erbrachten SW-Produktes ist maßgeblich von der Wahl des SW-Entwicklungsverfahrens und dessen technisch einwandfreier Anwendung abhängig. Es ist im Einzelfall zu untersuchen, ob Entwicklungsrisiken bestehen. Dabei sind nicht nur Eigenleistungen zu betrachten, sondern auch Leistungen von Partnerunternehmen, da die Gewährleistung für Fremdleistungen gegenüber dem Auftraggeber gleichermaßen besteht.



### RiGr9: Testverfahren

Testen und Verifikation sind neben Prävention und Prozess-Verbesserungen Möglichkeiten, eine hohe Qualität zu erreichen (vgl. YOURDON. 1992, S. 203). Wird nicht auf eine ausreichende Qualität geachtet, kann ein Projekt scheitern bevor die Ergebnisse verkauft werden können. Etwa 25%-30% der Zeit eines Designzyklus sollte für das Testen der erstellten Software eingeplant werden (vgl. FAIRLEY 1985, S. 61). Nicht-einheitliche Testverfahren im Netzwerk können Qualitätsrisiken nach sich ziehen.

### RiGr10: Mangelnde interne SW-Qualität (Wartbarkeit)

Die Wartbarkeit von Software ist ein Kriterium bei der Entwicklung von Software und zeigt an, mit welchem Aufwand und welchem Erfolg Änderungen in einem Systemzusammenhang von Applikationen durchgeführt werden können. Umso wichtiger wird diese Komponente ist, desto länger die geplante Nutzungsdauer der Software ist und desto geringer die Verfügbarkeit von Experten für das Sachgebiet ist. Dabei spielt z. B. die Dokumentation, insbesondere die exakte Spezifikation von Schnittstellen (Interfaces), wie auch ein modularer, stark gegliederter Aufbau (Zerlegung in elementare, einzeln testbare Einheiten) eine wichtige Rolle.

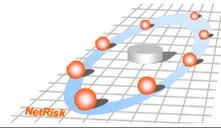
### RiGr11: Mangelnde Performance

Ein entscheidendes Qualitätskriterium eines SW-Produktes ist seine Performance. Dabei kann es sich um Übertragungsgeschwindigkeit, Echtzeitverhalten, Güteigenschaften, Verfügbarkeit, Durchsatz oder andere Leistungsparameter handeln. Es erscheint ersichtlich, dass eine mangelnde Performance ein hohes Risikopotential für den Erfolg des Projektes mit sich bringt.

### RiGr12: Mangelnde Usability

Usability oder zu Deutsch Gebrauchstauglichkeit bezeichnet die Eignung eines Produktes bei der Nutzung durch bestimmte Anwender in einem bestimmten Benutzungskontext, die vorgegebenen Ziele effektiv, effizient und zufriedenstellend zu erreichen. (vgl. DIN 2006 Teil 11) Ein SW-Produkt sollte auf das Benutzerklientel zugeschnitten sein, um mögliche Missverständnisse und Komplikationen frühzeitig zu beheben. Dabei gilt:

- Ein System ist angemessen, wenn es die zur Lösung der Arbeitsaufgabe erforderlichen Funktionen bereitstellt.



- Ein System ist handhabbar, wenn es dem Benutzer eine leichte Erlernbarkeit, Bedienbarkeit und Verständlichkeit ermöglicht.
- Ein System ist persönlichkeitsförderlich, wenn es den Fähigkeiten und Kenntnissen des Benutzers (Benutzermodell) angepasst ist und den Prinzipien der Arbeitsgestaltung entspricht.

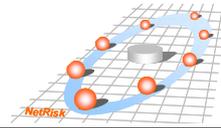
Einen Maßstab für die softwareergonomische Gestaltung bilden die Normen Teil 11 bis 17 sowie 110 der Normenreihe ISO 9241 „Ergonomische Anforderungen für Bürotätigkeiten mit Bildschirmgeräten“, die in den Jahren 1996 bis 1999 verabschiedet und anschließend sowohl als Europäische als auch als DIN-Normen übernommen worden sind. In Teil 110 „Grundsätze der Dialoggestaltung“ werden folgende Qualitätskriterien definiert:

1. Aufgabenangemessenheit – geeignete Funktionalität, Minimierung unnötiger Interaktionen
2. Selbstbeschreibungsfähigkeit – Verständlichkeit durch Hilfen / Rückmeldungen
3. Steuerbarkeit – Steuerung des Dialogs durch den Benutzer
4. Erwartungskonformität – Konsistenz, Anpassung an das Benutzermodell
5. Fehlertoleranz – erkannte Fehler verhindern nicht das Benutzerziel, unerkannte Fehler sollten leicht korrigierbar sein
6. Individualisierbarkeit – Anpassbarkeit an Benutzer und Arbeitskontext
7. Lernförderlichkeit – Anleitung des Benutzers, minimale Erlernzeit, Metaphern

### **Projektelement: Planungsqualität**

RiGr13: Projektplan

Die Planung legt vorausschauend fest, auf welchen Wegen, mit welchen Schritten, in welcher zeitlichen und sachlogischen Abfolge, unter welchen Rahmenbedingungen und mit welchen Kosten und Terminen ein Ziel erreicht werden soll (vgl. BALZERT 2000). Um Risiken zu reduzieren, beginnt jede erfolgreiche Software-Entwicklung mit einem verbindlichen Projekt-



plan, der vom Projektleiter erstellt wird. Ein Projektplan fasst die einzelnen Aufgaben zu Vorgängen zusammen; mehrere Vorgänge ergeben eine Phase. Meilensteine ermöglichen die Überwachung eines Projektes, sie kennzeichnen den Abschluss jeder Phase.

#### RiGr14: Entwurfs- und Ausführungsplanung

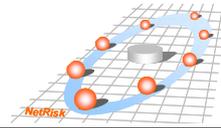
In der Softwareentwicklung wird für große Systeme zunächst ein Modell entworfen, welches die Daten und Abläufe des geplanten Systems repräsentiert. Anschließend wird dieses Modell in kleinere Bestandteile aufgeteilt und verwirklicht. Fehler in dieser Phase pflanzen sich gegebenenfalls weiter fort. In der Entwurfs- und Ausführungsplanung, die mit der Pflichtenhefterstellung abschließt, wird eine systemtechnische Lösung dargestellt. Hier werden die Hardware-Komponenten bestimmt und die Software-Spezifikationen festgelegt. Ein nicht vorhandenes Pflichtenheft ist laut Hertweck und Krcmar (HERTWECK, KRCMAR 1991, S. 3) ein Risiko für den gesamten SW-Entwicklungsprozess.

#### RiGr15: Technologieauswahl

Die Wahl der passenden Technologie für ein SW-Projekt ist eine mit weitreichenden Konsequenzen. Neben der Kompatibilität zu anderen Technologien sollte auch die Zukunftsfähigkeit des SW-Produktes beachtet werden. So kann es sich auszahlen, neuartige Technologien zu verwenden, die vielleicht zunächst höhere Kosten verursachen, aber das Produkt dafür auch flexibler und interessanter für den Markt machen. Auf der anderen Seite kann der Einsatz von unbekanntem Technologien zu einem erhöhten Risikopotential führen (KEIL et al. 1998, S.78)(STANDISH GROUP 1994, S.5). Darüber hinaus ist eine unpassende Technologiewahl einer der Hauptgründe für zunehmende Komplexität von SW-Entwicklungsprojekten und hat somit einen entscheidenden Einfluss auf den Projekterfolg (KOTULLA 2002).

#### RiGr16: Granularität der Planung

Durch eine zu grobe Planung kann es zu ungenauen Spezifikationen der Anforderungen kommen, was ein großes Risikopotential für den gesamten SW-Entwicklungsprozess mit sich bringt. Auf der anderen Seite kann z. B. das Vorhaben einer möglichst genauen Planung bereits in dieser Phase des Projektes zu Zeitverzögerungen führen.



### RiGr17: Flexibilität des Produktes

Ein SW-Produkt sollte auf sich ändernde Rahmenbedingungen flexibel reagieren können. Durch die Anpassung der vorhandenen Funktionen an neue Bedingungen und durch eine Erweiterbarkeit auf neue Aufgabengebiete wird eine längere Produktlebenszeit erreicht. Diese Flexibilität muss allerdings bereits bei der Entwicklung mit geplant und z. B. durch einen modularen, einfachen Aufbau unterstützt werden. Im Gegensatz dazu steht die Gefahr einer zu großen Komplexität aufgrund übermäßiger Flexibilität. Daraus resultiert bei steigender Komplexität ein Anwachsen des Wartungsaufwands.

### Hauptprojektelement: Projektzeit

Üblicherweise werden in einem Projekt zumindest für den Beginn und die Fertigstellung der Leistung oder von Teilen der Leistung Zeiten vereinbart. Die Nichteinhaltung ist für den Auftragnehmer mit Risiken verbunden.

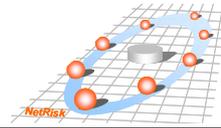
### Projektelement: Termine

#### RiGr18: Termine

Die Termintreue ist auch und gerade bei Unternehmensnetzwerken ein notwendiges Kriterium für eine erfolgreiche Partnerschaft. Unternehmenspartnerschaften, in denen diese Eigenschaft nicht gegeben ist, werden zwangsläufig in Verzug geraten und dadurch höhere Kosten für mehr Kapazitäten einplanen, eine Verschiebung des Fertigstellungstermins veranlassen oder gar einen Projektabbruch riskieren müssen.

#### RiGr19: Behinderungen

Behinderungen stellen Störungen dar, die den geplanten Projektverlauf, zumeist die Projekteschwindigkeit im negativen Sinne beeinflussen. Eine Frage in diesem Kontext ist die nach einer Fristenverlängerung bei nicht selbstverschuldeten Behinderungen; z. B. durch Änderungen während des laufenden Projektes durch den Auftraggeber.



## Projektelement: Arbeitsvorbereitung

### RiGr20: Ablauf- und Terminplanung

In der Ablauf- und Terminplanung wird der Ablauf der einzelnen Projektschritte so festgelegt, dass der vorgegebene Terminrahmen eingehalten wird. Im Unterschied zur Beurteilung der Terminrisiken ist an dieser Stelle nicht der Terminrahmen selbst auf Risiken hin zu untersuchen, sondern es sind die Voraussetzung für die Erstellung eines funktionsfähigen, wirkungsvollen Ablauf- und Terminplans zu beurteilen. Ein häufiger Mangel an Terminplänen ist das Fehlen von Pufferzeiten, welche allerdings erforderlich sind, da zumeist Abweichungen vom Ablaufplan während der Ausführung unvermeidbar sind und ohne entsprechende Pufferzeiten direkt zu Terminabweichungen führen.

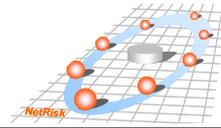
### RiGr21: Entwicklungsprozess

SW-Entwicklungsprozesse sind spezielle Vorgehensmodelle, die der Steuerung einer Softwareentwicklung von der Konzeption bis zum Einsatz im Echtbetrieb inklusive der im Echtbetrieb anfallenden Änderungen einer Software dienen. Ein effektiver Einsatz von SW-Entwicklungsprozessen ist laut Jones (JONES 1996) ein Erfolgsfaktor. Ungeeignete oder inkompatible SW-Entwicklungsprozesse in Netzwerken führen zu einem Entwicklungsrisiko.

### RiGr22: Bereitstellung Infrastruktur/ Personal

Voraussetzung für den Erfolg verteilter SW-Entwicklungsprozesse ist eine Infrastruktur, die die verteilte Entwicklung unterstützt. Darunter fallen z. B. Konferenzmöglichkeiten, ein Computernetzwerk und die passenden Werkzeuge (KOTULLA 2002, S.175).

Gleiches gilt für die personellen Kapazitäten, die für ein Projekt zur Verfügung gestellt werden. Bei nicht adäquater Besetzung des Projektteams ist ein termingerechter und somit erfolgreicher Abschluss des Projektes nur schwer zu erreichen. Besonders eine unklare Aufgabenstruktur birgt ein großes Risiko; z. B. wenn Mitarbeiter einem Projekt nicht vollständig zugeteilt sind oder durch andere Aufgaben abgelenkt werden (HERTWECK, KRCCMAR 1991, S.3).



## Hauptprojektelement: Beteiligte

Im Gegensatz zur klassischen Produktion fehlen im Softwareentwicklungsprozess ausführende Maschinen fast vollständig. Aus diesem Grund rücken neben dem Auftraggeber die direkt am Projekt arbeitenden Mitarbeiter in den Fokus. Aufgrund der Betrachtung von Unternehmensnetzwerken spielen natürlich auch Partnerunternehmen und deren Koordination eine gewichtige Rolle. Aus der herausragenden Stellung der Beteiligten lässt sich ableiten, dass sie natürlich auch ein großes Risikopotential bedeuten. Dieses wird im Folgenden näher beleuchtet.

### Projektelement: Beteiligte des Unternehmens/ Netzwerks

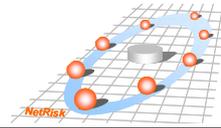
RiGr23: Mitarbeiter

Die beteiligten Personen und ihre Fähigkeiten in Bezug auf Teamarbeit tragen wesentlich zum Erfolg oder Misserfolg eines Projektes bei. Zu den allgemeinen Qualifikationen von Softwareentwicklern zählt Balzert (BALZERT 1998, S.140) dabei:

- Abstraktionsfähigkeit
- Sprachliche und schriftliche Kommunikationsfähigkeit
- Lernfähigkeit
- Wille zum lebenslangen Lernen
- Intellektuelle Flexibilität und Mobilität
- Kreativität
- Hohe Belastbarkeit
- Englisch lesen und sprechen

Eigenschaften, die z. B. für Mitarbeiter, die an Projekten unter internationaler Beteiligung mitarbeiten, sind laut Lichtenberger et al. (LICHTENBERGER et al. 1998, S.4):

- Strategisches Denken



Entscheidungen in einer internationalen Umgebung benötigen eine internationale Perspektive und Kenntnis internationaler Zusammenhänge.

- Internationale Soft Skills

Es wird eine gut ausgebildete Sensitivität, d. h. Wissen, Akzeptanz und Toleranz, gegenüber anderen Kulturen benötigt.

- Multifunktionelles Können und Erfahrung

Hierzu gehören technisches Wissen als auch die Fähigkeit, die ausländische Sprache zu sprechen.

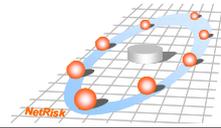
Mitarbeiter, die diesen Anforderungen nicht genügen, stellen ein Risiko für den Projekterfolg dar (BOEHM 1991, S.351).

#### RiGr24: Vorstand/ Management

Management umfasst alle Aktivitäten und Aufgaben, die von einem oder mehreren Managern durchgeführt werden, um die Aktivitäten von Mitarbeitern zu planen und zu kontrollieren, damit ein Ziel oder der Abschluss einer Aktivität erreicht wird, die durch die Mitarbeiter alleine nicht erreicht werden können (BALZERT 1998, S.6). Ein signifikantes Risiko kann durch die mangelhafte Unterstützung des Top-Managements auftreten (KEIL et al. 1998, S.78) (STANDISH GROUP 1994, S.5).

#### RiGr25: Projektleiter

Der Projektleiter hat die Verantwortung für den Projekterfolg inne und ist somit eine entscheidende Komponente für eben den Projekterfolg. Die Wahl eines fähigen und motivierten Projektleiters fällt dem Management zu. Diese Wahl beeinflusst das Projektrisiko in hohem Maße.



### **Projektelement: Schnittstellen**

#### RiGr26: Schnittstellenkoordination

Wenn verteilte Gruppen an einem gemeinsamen Softwareprojekt arbeiten, benötigt die Software eine sehr klare und gut durchdachte, stabile Architektur. Wenn die Software zu komplex verschachtelt ist, fehlt bei den beteiligten Gruppen die Übersicht; die gesamte Struktur ist unklar. Unklare Strukturen der einzelnen Bausteine bedeuten ein hohes Risiko für ein Entwicklungsprojekt (vgl. KOTULLA 2002, S.138). Verbessertes Schnittstellendesign ist unter anderem Aufgabe der Schnittstellenkoordination, die dafür sorgen soll, dass durch eine geschickte Aufteilung in Funktionsblöcke ein komplexes System für die einzelnen Teams relativ übersichtlich gemacht wird. Darüber hinaus sind firmenspezifische Schnittstellen und Verbindungen ein Risiko für einen SW-Entwicklungsprozess und können den zuvor beschriebenen Effekt noch verstärken (JONES 1996, S.64).

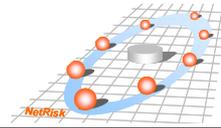
### **Projektelement: Auftraggeber/ Endbenutzer**

#### RiGr27: Auftraggeber

Die Spezifikationen des Auftraggebers sind maßgeblich für den Prozess der SW-Entwicklung. Aus diesem Grund sollte diese klar definiert und vor allem auch klar kommuniziert werden. Neben der inkompletten Spezifikation zu Beginn des Projektes sind die Änderungen der Spezifikation während des SW-Entwicklungsprozesses ein entscheidender Faktor für den Erfolg-/Misserfolg eines Projektes (vgl. STANDISH GROUP 1994).

#### RiGr28: Endbenutzer

Dem Endbenutzer sollte in einem SW-Entwicklungsprozess ein hohes Maß an Aufmerksamkeit gewidmet werden. Durch sein Urteil wird zum großen Teil entschieden, ob ein Produkt auf dem Markt zum Erfolg wird oder nicht. Besonders durch mangelnde Kommunikation zwischen Kunden und den Entwicklern können Missverständnisse und Fehleinschätzungen entstehen. Nach Churchman und Schainblatt (CHURCHMAN, SCHAINBLATT 1965), sowie Ives und Olson (IVES, OLSON 1984) sind ein gegenseitiges Verständnis von Kunde und Entwickler und die Mit-Einbeziehung von Endbenutzern in den Entwicklungsprozess wichtige Faktoren für eine erfolgreiche Softwareentwicklung.



## Projektelement: Beteiligte Unternehmen

### RiGr29: Kooperationen

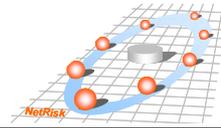
Die Zusammensetzung der Kooperationspartner birgt in sich bereits ein großes Risikopotential. Dabei kann sich die falsche Wahl der Partner auf jede Phase des Projektes auswirken. Neben eventuellen Wettbewerbssituationen und daraus resultierenden möglichen Informationsasymmetrien der Partner können Unterschiede in den Werkzeugen der zusammenarbeitenden Partner zu großen Problemen im Projekt führen.

### RiGr30: Partnerunternehmen

Es ist nicht selten der Fall, dass bei verteilten Projekten die Vorstellungen der Teams, wie das zu entwickelnde Produkt aussehen, wie das Ziel erreicht werden und wie der spätere Einsatz des Produktes verlaufen sollte, unterschiedlich sind. Ursache dafür sind, neben der schlechten Kommunikation zwischen den mitwirkenden Teams, vor allem die unterschiedlichen Prioritäten in den Entwicklungszentren. Je nach Zielvorstellung der Kooperationspartner können unterschiedliche Aspekte einer Software wichtig sein, und so hat jeder Partner ein Interesse daran, zuerst die von ihm am dringendsten benötigten Funktionen in der gemeinsamen Entwicklung voranzutreiben (KOTULLA 2002, S. 128). Mögliche unterschiedliche Zeitzonen erschweren oft die Synchronisation der einzelnen Arbeiten (MACHARZINA 1999). Darüber hinaus sollte die soziale Komponente, z. B. dem „Willen“, die zugrunde liegende Infrastruktur auch zum Informationsaustausch zu verwenden, aufgrund des enormen Risikopotentials in den Fokus des Interesses gesetzt werden (KOSTNER 1996, S.1671; KOPPENHÖFER et al. 1998, S.3). Als eine der erfolgreichsten Möglichkeiten „Vertrauen“ zwischen den Teams aufzubauen, werden „Face-to-Face“-Meetings genannt. Sie sollten zumindest am Anfang jedes Software-Projektes stattfinden.

### RiGr31: Kulturelle Unterschiede

Kulturelle Unterschiede spielen auch und besonders in einem international verteilten Softwareprojekt eine große Rolle, die zum Erfolg oder Misserfolg eines Projektes erheblich beitragen können. Sie führen oft unbewusst zu Missverständnissen oder ungünstigen Interpretationen, die eine komplexe Softwareentwicklung gefährden können. Missverständnisse, die durch kulturelle Unterschiede entstehen, bewirken meist eine Verschlechterung des Kommunikationsflusses der beteiligten Teams. Die daraus resultierende mangelhafte Kommunikation



wird oft als eine der Hauptgründe für das Scheitern von international verteilten Softwareprojekten beschrieben. Probleme in der Kommunikation zwischen Teams, die durch kulturelle Missverständnisse ausgelöst werden, sind nur sehr schwer zu identifizieren, da sich die meisten Beteiligten der Ursache für die Misstimmigkeiten selten bewusst sind.

Um Kommunikationsprobleme zwischen verschiedenen Kulturen zu verhindern, sollten bei den meisten verteilt aufgesetzten Projekten Entwickler zwischen beteiligten Niederlassungen ausgetauscht werden, die den lokal ansässigen Entwicklern ein Bild von Laboratorien in anderen Märkten vermitteln können und die Kommunikationswege in schwierigen Situationen offen halten (KOTULLA 2002, S.131ff). Ferner eignen sich kommunikationsintensive Projekte nur bedingt für eine standort- oder kulturübergreifende Entwicklung.

#### RiGr32: Wettbewerbsbeziehung

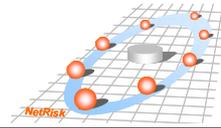
Einen nicht zu unterschätzenden Einfluss auf den Erfolg/ Misserfolg eines Projektes hat die Wettbewerbssituation der an einem Netzwerk beteiligten Unternehmen. Bei konkurrierenden Unternehmen steigt das Risiko opportunistischen Verhaltens der Projektpartner an. Informationsasymmetrien erschweren diese Problematik.

#### RiGr33: Räumliche Ausdehnung (Netzwerkreichweite)

Die größten Probleme in verteilten Projektteams tauchen in den Bereichen Synchronisation und Kommunikation auf. Gerybadze, Meyer-Krahmer und Reger (GERYBADZE, MEYER-KRAHMER, REGER 1997) schreiben, dass standortübergreifende Teambildung nicht einfach, interkulturell zusammengesetzte Projektarbeit jedoch oftmals noch schwieriger ist. Daraus lässt sich ableiten, dass das Risiko für ein verteiltes Softwareprojekt mit der Ausdehnung des Netzwerkes, von lokal, über regional und national, bis hin zu international ansteigt.

#### RiGr34: Anzahl der Netzwerkpartner

Die Anzahl der Netzwerkpartner kann beliebig variieren. Besonderes Augenmerk sollte dabei auf die Schnittstellenkoordination (s.o.) gelegt werden. Bei wachsender Anzahl der beteiligten Unternehmen steigt auch der Bedarf an Koordination der Schnittstellen und somit das Risikopotential.



### RiGr35: Vertragliche und rechtliche Eigenschaften

Ein Unternehmensnetzwerk kann auf verschiedene Arten konzipiert werden. In diesem Kontext spielt auch die Netzwerkgrenze eine gewichtige Rolle. Dabei sind Formen wie offen, permeabel und geschlossen denkbar. Die andere Frage ist die nach der Verbindlichkeit. Basiert das Netzwerk entweder auf unverbindlichen Absprachen, also vertrauensbasiert oder auf einem verbindlichen Vertrag? Weitere Mischformen sind hier ebenso zu konstruieren. Auf das Risiko haben diese Entscheidungen einen signifikanten Einfluss. Netzwerke mit einer offenen Grenze erhöhen bspw. die Gefahr eines Know-how-Abflusses, da im Netzwerk beigestelltes Wissen später weiteren hinzugetretenen Unternehmen zugänglich ist. Die vertraglichen Risiken steigen ebenfalls, da die Komplexität des Regelwerkes oft mit der Netzwerkkomplexität zunimmt. Diese Risiken können allerdings nur von Juristen fundiert eingeschätzt und beschrieben werden.

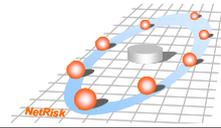
### RiGr36: Zeitliche Determinanten

Geographisch verteilte Teams benötigen vergleichsweise lange, bis eine gute Zusammenarbeit erreicht ist (HERN 1996). Dabei ist es im Vergleich zu persönlichen Treffen viel schwieriger, Vertrauen oder Teamgeist am Telefon oder über Online-Medien zu entwickeln. Vertrauen aller Beteiligten ineinander wird allerdings als essentiell für einen erfolgreichen Projektabschluss genannt (KOSTNER 1996) (KOPPENHÖFER et al. 1998). Dieses Vertrauen wird durch eine langfristige Partnerschaft aufgebaut. Dabei ist davon auszugehen, dass nicht funktionierende Partnerschaften frühzeitig beendet werden. Somit birgt eine langfristige Partnerschaft ein geringeres Risiko in sich als eine kurzfristige Partnerschaft.

## **Projektelement: Ressourcen**

### RiGr37: Personalressourcen

Aufgrund dynamischer Entwicklungen, müssen die Projektpartner oft nicht nur eine ausreichende „Anfangsausstattung“ an fähigen und motivierten Mitarbeitern einbinden, sondern sie müssen oft auch weiteres Personal kurzfristig bei Engpässen bereitstellen oder das bestehende Team zu Mehrarbeit verpflichten können, so dass akute Probleme kurzfristig gelöst werden. Unzureichende Personalressourcen führen entweder zu Zeit- oder Qualitätsproblemen.



## Hauptprojektelement: IT-Strukturen

Die bestehenden IT-Strukturen umfassen die Kommunikationsinfrastruktur, Entwicklungswerkzeuge (z. B. Datenbanksystem oder Entwicklungsumgebung) und auch bereits entwickelte, getestete sowie am Markt platzierte SW-Module.

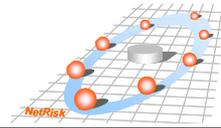
### RiGr38: Kommunikationsinfrastruktur

Bei Kostner (KOSTNER 1996, S.65ff) wird mangelnde oder schlechte Kommunikation als die häufigste Ursache für das Scheitern von Softwareprojekten benannt. Mangelnde Kommunikation ist oft Ausgangspunkt vieler weiterer Probleme, die sich im späteren Verlauf eines Projektes kaum noch beheben lassen. Stimmt die Kommunikation und damit auch die Koordination in und zwischen den einzelnen an einem Projekt arbeitenden Teams nicht, kann weder die Planung noch die Kontrolle funktionieren. Es kann vorkommen, dass die Anforderungen der Software für einige Teams unklar bleiben oder falsch verstanden werden. Notwendig aber nicht hinreichend für eine funktionierende Kommunikation ist eine geeignete Kommunikationsinfrastruktur. Ohne funktionierende Computernetzwerke, Telefonverbindungen, Telekonferenzmöglichkeiten, E-Mail und Werkzeugen für verteilte Gruppenarbeit wie z. B. Datenbanken, Terminplansysteme, verteilte Bibliotheken für Code und Dokumentation ist eine globale Entwicklung von Software von unmöglich. Hertweck und Krcmar (HERTWECK, KRCMAR 1991) nennen drei Punkte, warum virtuelle Kommunikation und SW-Projekte miteinander zusammenhängen:

1. Die Produktion von Software und Systemen erfordert ein hohes Maß an verbaler, schriftlicher und visueller Kommunikation im Verlauf des Erstellungsprozesses.
2. Die Mittel, mit denen kommuniziert wird, sind wie in kaum einer anderen Branche elektronischer und meist verteilter Art (E-Mail, gemeinsame Projektdatenbanken).
3. Oft sind Projektteams auf verschiedene Standorte verteilt und vereinen Mitglieder unterschiedlicher Organisationen.

### RiGr39: Bestehende SW-Module

Bereits entwickelte, getestete und in der Praxis erprobte SW-Module bieten die Möglichkeit, Entwicklungszeit und monetäre Mittel zu begrenzen. Laut Jones (JONES 1996) gibt es eine



grundlegende Regel für Softwareprojekte: Es gibt unendlich viele Gründe, warum ein komplexes Softwareentwicklungsprojekt scheitern kann, es gibt jedoch nur sehr wenige Wege, wie ein solches Projekt erfolgreich beendet werden kann. Dabei ist die Wiederverwendung von Software ein essentieller Erfolgsfaktor. Je weniger auf etablierte Module zurückgegriffen werden kann, je größer wird das Entwicklungsrisiko.

#### RiGr40: Werkzeuge

Global einheitliche Entwicklungswerkzeuge sind für die effektive Arbeit in verteilten Teams ein zentraler Erfolgsfaktor. Im Gegensatz zu einer lokalen Entwicklung an einem Ort kann es bei verteilten Teams öfter vorkommen, dass unterschiedliche Werkzeuge eingesetzt werden. Dazu gehören u. a. Compiler, Editoren, Code Datenbanken, Debugger und Textverarbeitungssysteme. In diesem Fall können einige Aktivitäten nicht zufriedenstellend oder nur mit wesentlich höherem Zeitaufwand gelöst werden. Idealerweise hat deshalb jedes Team einer verteilten Entwicklungsumgebung eine gemeinsame Werkzeugbasis. Abweichungen von dieser Regel sollten begründet sein.

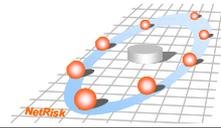
Die Teams müssen Zugriff auf dieselben Codedatenbanken besitzen und die Testlabore sollten jeweils gleich ausgestattet sein, da andernfalls Ergebnisse oder Tests nicht an anderen Lokationen nachvollzogen werden können. Darüber hinaus gehören auch CASE-Tools, zentrale Server, eine verteilte Softwarebibliothek, Versionsverwaltungssysteme und Tools, die einen Softwareload generieren können, zum einheitlichen Toolset (KOTULLA 2002, S.157f).

#### RiGr41: Verlässlichkeit der Infrastruktur

Die IT-Infrastruktur muss verlässlich sein. Dabei spielt nicht nur die interne Infrastruktur, z. B. Netzwerk oder Server, eine Rolle, sondern auch die Externe, wie z. B. Stromversorgung oder Telefonnetz. Gerade in Schwellenländern ist diese Verlässlichkeit nicht immer gegeben und kann bei starken Störungen, neben Problemen im Entwicklungsprozess vor Ort, auch zu einem großen Kommunikationsrisiko bei verteilter Entwicklung führen.

### **Hauptprojektelement: Gesetze und Vorschriften**

Auch für verteilte Projekte gelten Gesetze und Vorschriften; z. B. den Schutz des geistigen Eigentums. Durch die verteilte Entwicklung in mehreren Ländern sind diese Projekte allerdings mit verschiedenen juristischen Fragestellungen konfrontiert, die sich aus den unter-



schiedlichen Auslegungen in den verschiedenen Gesetzesbüchern ergeben. Getreu dem Motto „Andere Länder, andere Sitten“ ist der o.g. Schutz des geistigen Eigentums nicht immer per Gesetz gegeben bzw. werden Verstöße nicht konsequent verfolgt. Eine naive Herangehensweise an diese oder ähnlich geartete Probleme kann schnell zu Risiken führen.

#### RiGr42: Copy Rights

Ein wichtiges und gerade in der SW-Branche häufig entscheidendes Thema ist jenes des „Copy Rights“. Dabei werden im angloamerikanischen Rechtssystem (im Gegensatz zum kontinentaleuropäischen Urheberrecht) die Entscheidungs- und Verwertungsrechte über ein Werk/Produkt oft nicht dem Urheber (zum Beispiel dem Entwickler) zugestanden, sondern den wirtschaftlichen Rechteinhabern. Gerade bei verteilter Entwicklung mit mehreren beteiligten Unternehmen ist diese Frage häufig nicht ohne weiteres zu beantworten. Die Risikolage sollte von juristischem Personal geprüft werden.

#### RiGr43: Lizenzen (z. B. general public licence)

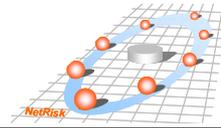
Lizenzen müssen frühzeitig gesichert werden und sollten vor Beginn des Projektes genauestens überprüft werden. Dieses Feld birgt gerade in der SW-Entwicklung ein hohes Risikopotential. Unter dem Begriff „general public licence“ versteht man die Verpflichtung bei Nutzung von „open source“ Programmen, das Endprodukt auch wieder kostenfrei zur Verfügung zu stellen.

#### RiGr44: Sonstige Gesetze und Vorschriften

Neben den bereits beschriebenen Punkten müssen sich die Verantwortlichen auch unterschiedlicher Arbeitsgesetze bewusst sein, die an den einzelnen Standorten eines verteilten Teams gelten (MACHARZINA 1999). Zu den sonstigen Gesetzen und Vorschriften sind darüber hinaus auch z. B. Arbeitnehmerrechte oder Datenschutzbestimmungen zu zählen, die ein Risiko für die erfolgreiche Abwicklung des Projektes bedeuten können.

### **Hauptprojektelement: Vergütungssoll**

Das Vergütungssoll ist die im Vertrag festgelegte Entlohnung für eine fehlerfreie und im Zeitplan erbrachte Leistung. Dieser Ausgang eines Projektes ist allerdings nur in knapp einem Drittel der Software-Entwicklungsprojekte zu beobachten (vgl. STANDISH GROUP 2004).



Darüber hinaus beinhaltet die Kostenkalkulation an sich bereits ein großes Risikopotential. Die Unsicherheit in der Aufwands- und Kostenabschätzung ist bei Softwareprojekten im Allgemeinen größer als bei herkömmlichen Projekten (vgl. FEYHL, FEYHL 1996).

#### RiGr45: Unklares Vergütungssoll

Die Vergütung durch den Auftraggeber ist durch den Vertrag bestimmt. Fraglich sind zumeist nur Auswirkungen von Verspätungen und Abweichungen vom Leistungssoll. Sind diese nur unzureichend oder lückenhaft im Vertrag konkretisiert, kann es zu Problemen bei der Berechnung des Vergütungssolls und ggf. zu Zahlungsausfällen kommen.

### **Hauptprojektelement: Preisermittlung**

Die Risiken der Preisermittlung resultieren zum einen aus der Kostenermittlung und zum anderen aus der Preisbildung. Eine effektive Kostenermittlung ist für Jones (JONES 1996) ein Erfolgsfaktor für SW-Entwicklungsprojekte.

#### **Projektelement: Kalkulation allgemein**

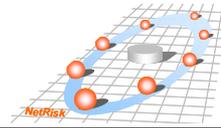
##### RiGr46: Kalkulation allgemein

In der Kalkulation ist die Leistung mit den bei ihrer Erstellung anfallenden Kosten zu bewerten. Dabei ist es wichtig, die einzelnen Elemente korrekt in die Kalkulation mit einfließen zu lassen. So kann es zu einem nicht kostendeckenden Angebotspreis kommen, wenn Teile der Leistung nicht berücksichtigt werden. Im Gegensatz dazu kann es zu einem erhöhten Angebotspreis kommen, wenn Bestandteile der Entwicklung mehrfach Berücksichtigung finden. Das Risiko der Kalkulation wird generell durch einen zu geringen Ansatz von Wagnis und Gewinn erhöht.

#### **Projektelement: Einzelkosten**

##### RiGr47: Kalkulation Lohnkosten

Die Lohnkosten sind prinzipiell eine gut vorhersehbare Größe, wenn keine Probleme und Verzögerungen und damit weitere und zum Teil besser ausgebildete Mitarbeiter in das Projekt mit eingebunden werden müssen. Da, wie bereits von der Standish Group (STANDISH



GROUP 2004) erwähnt, dieses Szenario nicht unwahrscheinlich ist, ist auch die Kalkulation der Lohnkosten grundsätzlich mit einem Risiko behaftet.

#### RiGr48: Kalkulation Infrastrukturkosten

Die Infrastrukturkosten können bei verteilter Softwareentwicklung aufgrund der verschiedenen Voraussetzungen und möglichen Investitionen zur Anpassung der Infrastrukturen ein Risiko für die Kalkulation und somit auch für den Projekterfolg bedeuten.

#### RiGr49: Kalkulation sonstiger Kosten

Ebenfalls sollten die finanziellen Aufwendungen für den eventuellen Erwerb von z. B. Sekundärlizenzen oder notwendiger externer Ressourcen einkalkuliert werden.

### **Projektelement: Gemeinkosten**

#### RiGr50: Kalkulation Gemeinkosten des Projektes

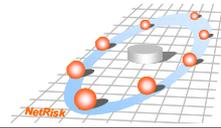
Gemeinkosten sind Kosten, die einem Kostenträger (z. B. verkaufsfähiges Produkt oder Dienstleistung) nicht direkt zugerechnet werden können. So können z. B. erhöhte Kosten für die Wartung der Software, an der viele Unternehmen mitgearbeitet haben, häufig nicht einem Unternehmen in Rechnung gestellt werden. Da allerdings gerade diese Wartungskosten laut Prakken (PRAKKEN 1997) knapp zwei Drittel der Kosten eines Softwareentwicklungsprojektes ausmachen, wird das enorme Risikopotential auf diesem Gebiet klar.

### **Projektelement: Angebotspreis**

#### RiGr51: Festlegung des Angebotspreises

Im Rahmen der Preisbildung wird auf Grundlage der Kalkulation der Angebotspreis festgelegt. Aus geschäfts- und wettbewerbspolitischen Gründen kann es dazu kommen, dass dieser vom kalkulatorisch ermittelten Preis erheblich abweicht. Gründe hierfür können sein:

- Beschäftigungszwang
- Vermutete Chance zur Aufbesserung des Preises
- Erwirtschaftung eines Deckungsbeitrages



### **Hauptprojektelement: Zahlung**

Die Zahlung der Vergütung ist Voraussetzung für die Sicherstellung der Liquidität des Unternehmens. Liquidität bezeichnet die Fähigkeit, allen Zahlungsverpflichtungen nachkommen zu können. Diese wird sowohl durch Zahlungsverzögerungen bzw. Zahlungsausfällen oder auch Vertragsstrafen, welche nachfolgend genauer beschrieben werden, gefährdet.

#### **RiGr52: Zahlungsverzögerungen, Zahlungsausfall**

Das Risiko der Illiquidität ist nur Unternehmensebene in einer Gesamtbetrachtung der zahlungsrelevanten Vorgänge zu beurteilen. Somit kann im Rahmen der Betrachtung von Projektrisiken nur das Verlustrisiko betrachtet werden, welches sich aus verspäteten Zahlungen oder kompletten Zahlungsausfällen ergibt. Dieses Risiko ist selten hundertprozentig zu eliminieren.

#### **RiGr53: Vertragsstrafen**

Für den Fall einer nicht vertragsgemäßen Leistungserfüllung können Vertragsstrafen vereinbart werden. So können z. B. bei einer nicht vollständigen Erfüllung des festgelegten Leistungsumfangs, Kosten entstehen, die bei der Projektplanung nicht einkalkuliert waren.