

## Der NetRiskManager

### 7.1.1 Fachlicher Rahmen

Software-Entwicklungsprozesse in einem Unternehmen sind mit großen Risikopotentialen behaftet. Für den Fall von Software-Entwicklungsprozessen in Unternehmens-Netzwerken nehmen diese aufgrund der Kooperationen noch zu. In der frühzeitigen Erkennung und Bewertung dieser Risiken liegt der Ansatzpunkt des Projektes „NetRisk“. Dafür wird eine Software entwickelt, die basierend auf den Attributen der beteiligten Unternehmen, sowie des zu betrachtenden Projektes eine Liste von potentiellen Risiken des Projektes berechnet. Hierzu sind definierte, kausale Zusammenhänge notwendig, die zum einen die erwähnten Attribute mit den Risiken und diese wiederum mit den entsprechenden Maßnahmen verknüpfen. Diese Maßnahmen werden zu den jeweiligen Risiken ausgegeben und ermöglichen somit eine frühzeitige Behebung der vom Projekt- und Risikoverantwortlichen eines Unternehmens, nachfolgend vereinfachend *User* genannt, als relevant angesehenen Risiken.

Das Programm bietet dem *User* verschiedenste Möglichkeiten. Er kann Unternehmen und nach Absprache mit den Verantwortlichen der beteiligten Unternehmen ein Projekt anlegen. Dafür, wie auch für die Teilnahme an bereits existierenden Projekten, muss die Projektbeteiligung des Unternehmens erstellt werden. In ihr sind die projektspezifischen Attribute der Unternehmung zusammengefasst. Beispielhaft seien hierfür die Programmiersprache oder die vom Unternehmen abgestellten Mitarbeiter für das Projekt genannt.

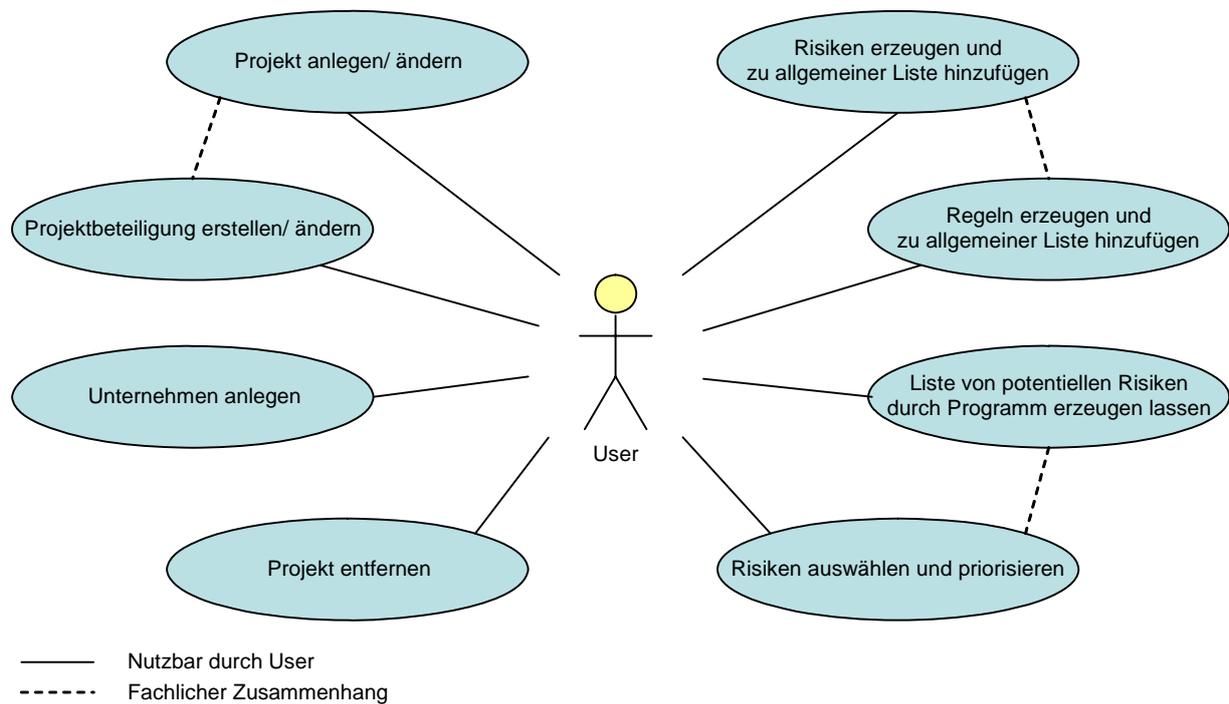
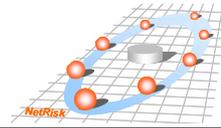
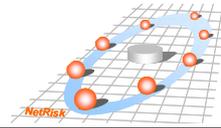


Abbildung 5-14: Use-Case-Diagramm für den NetRiskManager

Darüber hinaus betrachtet der *User* die Risiken eines Projektes und ihre Auswirkungen auf die Zusammenarbeit innerhalb des Projektes. Hierzu kann er noch nicht vorhandene Regeln, Risiken und Verknüpfungen zwischen den Regeln und Risiken zu bereits bestehenden Datensätzen hinzufügen. So kann für eine neue Regel, deren Risiken schon existieren, eine neue Verknüpfung einen kausalen Zusammenhang ausdrücken. Regeln basieren auf den Attributen der Unternehmung, der Projektbeteiligung und des eigentlichen Projektes. Diese Elemente werden logisch miteinander verknüpft. Falls die Regel erfüllt ist, folgt daraus eine Risikomenge. Dies impliziert einen Zusammenhang zwischen den Datensätzen der Risiken und Regeln. Darauf basierend kann das Programm, wie oben bereits erläutert, eine Liste von potentiellen Risiken für das gewählte Projekt erzeugen. Die einzelnen Risiken werden vom *User* bewertet und priorisiert. Diese Bewertung bildet die Basis für Änderungsvorschläge für das Projekt oder die Projektbeteiligung. Durch die kontinuierliche Begleitung des Projektes ist es dem *User* nach Absprache mit den beteiligten Unternehmen möglich, sowohl das Projekt an sich, wie auch die Projektbeteiligung seines Unternehmens zu ändern. Nach Vollendung oder Abbruch eines Projektes kann der *User* das Projekt entfernen.



### 7.1.2 Struktur: Ursachen und Wirkungen für Risiken beim SW-Entwicklungsprozess

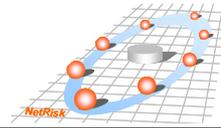
Den drei Objekten *Unternehmen*, *Projektbeteiligung* und *Projekt* kommt im Rahmen des „NetRisk“-Projektes eine große Bedeutung zu. Diesen Objekten sind verschiedene Eigenschaften zugeordnet.

Company	ProjectParticipation	Project
E1: softwareModel: string E2: groupware: string E3: projectMgtSW: string E4: configurationsMgtSW: string E5: timeZone: int E6: testingStrategy: int E7: cmmi: int E8: spice: int E9: rating: int E10: countryStability: int E11: budget: int E12: numberEmployers: int	E1: developmentEnviroment: string E2: frameworks: string E3: programmingLanguage: string E4: databaseSystem: string E5: language: string E6: bugTracker: string E7: taskExperience: float E8: motivation: int E9: comunicationClient: int E10: testerRate: float E11: automatedTesting: boolean E12: numberEmployers : int E13: budget: int E14: timeFrame: int	E1: parallelDevelopment: boolean E2: competition: boolean E3: continuousPartnership: boolean E4: budget: int E5: timeFrame: int

Abb. 5-15: Objekt-Diagramm des NetRiskManager

Nachfolgend sind jene Eigenschaften aufgelistet, die den *Unternehmen* C1, C2, ..., Cm zugeordnet werden. Sie werden für alle Projekte des Unternehmens als konstant angesehen:

- *Entwicklungsprozess* (softwareModel: string) XP, Wasserfall, RUP, ...  
 dient zur Steuerung einer Softwareentwicklung von der Konzeption bis zum Einsatz im Echtbetrieb.
- *Groupware* (groupware: string) notes, exchange, ...  
 bezeichnet eine Software zur Unterstützung der Zusammenarbeit in einer Gruppe über zeitliche und/oder räumliche Distanz hinweg.
- *Software-Test* (testingStrategy: int) 0, 1, 2, 3  
 bezeichnet man in der Informatik ein mögliches Verfahren zur teilweisen Verifikation und Validierung eines Programms. Diese Verfahren werden nach Informationsstand in



zwei Klassen eingeteilt: white- und black-box-Tests. Bei white-box-Tests werden die Tests mit Kenntnissen über die innere Funktionsweise des zu testenden Systems entwickelt. Bei black-box-Test wird diese innere Funktionsweise nicht beachtet. Man beschränkt sich stattdessen auf funktionsorientiertes Testen. 0 wird angegeben, falls keines der angesprochenen Testverfahren angewendet wird. Bei alleinigem Testen durch white-box oder black-box wird der Variable 1 bzw. 2 zugeordnet. Wenn beide Testverfahren durchlaufen werden, wird dies durch den Wert 3 gekennzeichnet.

- *Anzahl Mitarbeiter* (numberEmployers: int)

beschreibt wie viele Mitarbeiter das Unternehmen hat.

- *CMMI* (cmmi: int) 1, 2, ..., 5

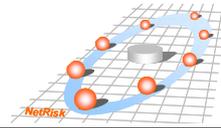
ist ein Prozessmodell zur Beurteilung und Verbesserung der Qualität ("Reife") von Produkt-Entwicklungsprozessen in Organisationen. Ein solches Prozessmodell kann verwendet werden, um die Stärken und Schwächen einer Produktentwicklung objektiv zu analysieren, um Verbesserungsmaßnahmen zu bestimmen und diese in eine sinnvolle Reihenfolge zu bringen.

- *SPICE* (spice: int) 1, 2, ..., 6

oder ISO 15504 (vgl. ISO 15504-1 2004, ISO 15504-2 2003, ISO 15504-3 2004, ISO 15504-4 2004, ISO 15504-5 2006) stellt ein Modell für das Assessment von Unternehmensprozessen dar. Die Verbesserung von Prozessen (Process Improvement) einerseits und die Bestimmung des Prozessreifegrads (Capability Determination) andererseits bilden die Kernpunkte dieses Modells.

- *Rating* (rating: int) 1, 2, ..., 5

bezeichnet die Bewertung von Anleiheemittenten im Hinblick auf den Gläubigerschutz. Dabei wird die Fähigkeit und der Willen des Gläubigers bewertet, seine Verpflichtungen rechtzeitig zu begleichen. Der User bewertet die Unternehmen anhand der Veröffentlichungen der Rating Agenturen (Standard & Poor's, Moody's, Fitch, ...). Diese werden im Rahmen des „Basel II“ Abkommens in 5 Kategorien eingeteilt. Dadurch ergibt sich eine Einteilung von 1 (nicht gerated) bis 5 (gut) basierend auf „Basel II“.



- *Zeitzone* (timeZone: int) -12/ +12

ist ein Abschnitt der Erdoberfläche, auf dem eine gemeinsame Uhrzeit gilt. Angegeben wird die Zeitdifferenz zur „Greenwich Mean Time“ (GMT).

- *Projektmanagement/SW* (projectMgtSW: string) Excel, Project, eS-MART,...

beschreibt alle organisatorischen Verfahren und Techniken, die mit der erfolgreichen Abwicklung eines Projektes verbunden sind. Hierfür sind zahlreiche Software-Lösungen vorhanden.

- *Umsatz* (budget: int)

dient zur Einschätzung der Größe und finanziellen Leistungsfähigkeit eines Unternehmens und wird in Euro angegeben.

- *KonfigurationsMgt.* (configurationMgtSW: int) CVS, SourceSave, ...

ist eine Managementdisziplin, die organisatorische und verhaltensmäßige Regeln auf den Produktlebenslauf einer Konfigurationseinheit von seiner Entwicklung über Herstellung und Betreuung anwendet.

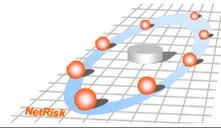
- *Landesstabilität* (countryStability: int) 1, 2, ... , 5

gibt die relative Gefährdung eines Unternehmens im Bezug auf den Standort an. Beispielhaft seien hierfür Rechtssicherheit, Korruption oder politische Stabilität. Die Bewertung erfolgt auf Basis einer entsprechenden Liste der CIA (vgl. CIA 2006). Eine endgültige Einstufung muss aber vom User erbracht werden.

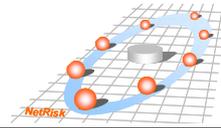
Eigenschaften des Unternehmens, die von Projekt zu Projekt variieren können, werden unter dem Objekt *Projektbeteiligung* (C1.Pp, C2.Pp, ... Cm.Pp) zusammengefasst:

- *Entwicklungsumgebung* (developmentEnviroment: string) Visual Studio, Eclipse,...

ist ein Anwendungsprogramm zur Entwicklung von Software.



- *Frameworks* (frameworks: string) .NET, J2EE,..  
werden insbesondere im Rahmen der objektorientierten Softwareentwicklung sowie bei komponentenbasierten Entwicklungsansätzen verwendet. Ein Framework gibt in der Regel eine Anwendungsarchitektur vor.
- *Programmiersprache* (programmingLanguage: string) Java, C++, C#,...  
wird zur Entwicklung eines Projektes verwendet.
- *Datenbanksystem* (databaseSystem: string) Access, Oracle, SQL,...  
ist ein System zur elektronischen Datenverwaltung. Die wesentliche Aufgabe eines DBS ist es, große Datenmengen sicher zu speichern und für Abfragen durch Benutzer oder Anwendungssoftware bereitzustellen.
- *Sprache* (language: string) Deutsch, Englisch,...  
gibt an, über welche Sprachkenntnisse die Mitarbeiter eines Projektes in dem Unternehmen verfügen.
- *Anzahl Mitarbeiter* (numberEmployers : int)  
beschreibt wie viele Mitarbeiter des Unternehmens in einem speziellen Projekt beschäftigt sind.
- *Bug Tracker* (bugTracker: string) Track+, Bugzilla,...  
sind in der Software-Entwicklung eingesetzte Computerprogramme, die als Werkzeug der Erfassung und Dokumentation von Programmfehlern dienen.
- *Testmethode* (automatedTesting: boolean)  
zeigt, ob die Software automatisiert getestet wird. Dabei wird davon ausgegangen, dass manuelle Testverfahren zum Standard gehören.



- *Anteil Tester* (testerRate: float)

im Vergleich mit der Anzahl der Entwickler des Projektes (Tester/Entwickler).

- *Berufserfahrung* (taskExperience: float)

der Mitarbeiter mit der Aufgabenstellung. Dabei wird auf die durchschnittliche Erfahrung der Mitarbeiter mit gleich gearteten Projekten in Jahren abgestellt.

- *Motivation* (motivation: int) 1, 2, ..., 5

der Mitarbeiter im Rahmen des Projektes.

- *Kundenkommunikation* (comunicationClient: int) 1, 2, ..., 5

bezeichnet die Kommunikationsbeziehung zwischen dem Kunden und den Projektverantwortlichen. Darunter versteht man zum einen den Austausch in der Planung des Projektes, aber auch die projektbegleitende Kommunikation.

- *Budget* (budget: int)

gibt die finanzielle Ausstattung seitens des Unternehmens in Euro an.

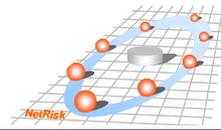
- *Zeitraumen* (timeFrame: int)

gibt Auskunft über den geplanten Zeitraum zur Vollendung des unternehmensinternen Projektes.

Dabei werden die Eigenschaften Kommunikationsgüte mit dem Kunden und die Motivation der Mitarbeiter vom User betrachtet und mit einem Wert zwischen 1 (schlecht) und 5 (gut) bewertet.

Eigenschaften des Projektes (P) werden unter dem Objekt *Projekt* zusammengefasst:

- *paralleles Arbeiten* (parallelDevelopment: boolean)



zielt auf die eigentliche Entwicklung des Projektes ab. Entweder entwickeln die Partner parallel oder sequentiell. Unter Zweitem versteht man, dass die Projektpartner auf das Ergebnis von anderen Projektpartnern angewiesen sind.

- *Wettbewerb* (competition: boolean)

gibt die Beziehung zwischen den beteiligten Unternehmen im Bezug auf ihre Konkurrenzsituation an. Informationsasymmetrien werden durch vorhandenen Wettbewerb verstärkt.

- *feste Partnerschaft* (continuousPartnership: boolean)

beschreibt, ob die Projektpartner bereits in der Vergangenheit mehrere Projekte miteinander durchgeführt haben. Dadurch können scheinbare Probleme bereits im Rahmen früherer Projekte gelöst worden sein.

- *Budget* (budget: int)

gibt die finanzielle Ausstattung des Projektes in Euro an.

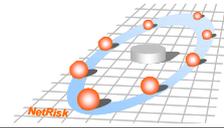
- *Zeitraumen* (timeFrame: int)

gibt Auskunft über den geplanten Zeitraum zur Vollendung des Gesamtprojektes.

Diese o.g. Attribute fließen in die Konstruktion der SimpleRules ein (Beispiele siehe Abb.3).

Company		ProjectParticipation		Project	
S_C1	C1.E1 != C2.E1	S_Pp1	C1.Pp.E1 != C2.Pp.E1	S_P1	P.E1 == true
S_C2	C1.E2 != C2.E2	S_Pp2	C1.Pp.E2 != C2.Pp.E2	S_P2	P.E2 == false
S_C3	C1.E3 != C2.E3	S_Pp3	C1.Pp.E3 != C2.Pp.E3	S_P3	P.E3 == false
S_C4	C1.E4 != C2.E4	S_Pp4	C1.Pp.E4 != C2.Pp.E4		
S_C5	C1.E5 - C2.E5 >=8	S_Pp5	C1.Pp.E5 != C2.Pp.E5		
S_C6	C1.E6 < 2	S_Pp6	C1.Pp.E6 != C2.Pp.E6		
S_C7	C1.E7 < 3	S_Pp7	C1.Pp.E7 < 3		
S_C8	C1.E8 < 3	S_Pp8	C1.Pp.E8 < 3		
S_C9	C1.E9 < 3	S_Pp9	C1.Pp.E9 < 3		
S_C10	C1.E10 < 3	S_Pp10	C1.Pp.E10 < 0,2		
		S_Pp11	C1.Pp.E11 == false		

Abbildung 5-16: Beispiele für SimpleRules



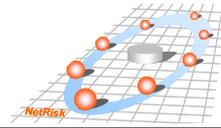
ComplexRules			
Cx1	S_P3	S_C5	
Cx2	S_P3	S_C7	
Cx3	S_P3	S_C8	
Cx4	S_P3	S_C9	
Cx5	S_P3	S_Pp5	
Cx6	S_P3	S_P2	
Cx7	S_P3	S_P1	S_C1
Cx8	S_P3	S_P1	S_C2
Cx9	S_P3	S_P1	S_C3
Cx10	S_P3	S_P1	S_C4
Cx11	S_P3	S_P1	S_Pp1
Cx12	S_P3	S_P1	S_Pp2
Cx13	S_P3	S_P1	S_Pp3
Cx14	S_P3	S_P1	S_Pp4
Cx15	S_P3	S_P1	S_Pp6
Cx16	S_C6		
Cx17	S_C7		
Cx18	S_C8		
Cx19	S_C9		
Cx20	S_C10		
Cx21	S_Pp10		
Cx22	S_Pp11		
Cx23	S_Pp12		
Cx24	S_Pp13		
Cx25	S_Pp14		

Abb. 5-17: Beispiele für ComplexRules

Sie werden zum einen durch den Vergleich der Unternehmen oder Projektbeteiligungen untereinander (z. B. S\_C1), durch den Vergleich mit einer Konstanten (z. B. S\_C7) oder durch arithmetische Operationen zwischen den Integer-Attributen (z. B. S\_C5) gebildet. Dabei werden die SimpleRules paarweise durchlaufen, um bei einem potentiellen Risiko „true“ auszugeben. Basierend auf diesen SimpleRules sind die ComplexRules konstruiert (Beispiele siehe Abb.4). Dabei können mehrere SimpleRules logisch durch „und“ miteinander verknüpft werden.

Ein Beispiel für einen solchen Prozess bis hin zur Angabe eines potentiellen Risikos wäre Folgendes:

Basierend auf dem Attribut timeZone (E5) der Unternehmen C1 und C2 wird die SimpleRule  $C1.timeZone - C2.timeZone \geq 8$  (S\_C5) erzeugt. Diese sagt bei „true“ aus, dass sich durch die verschiedenen Zeitzonen, in denen die Projektpartner arbeiten ein potentielles Risiko ergibt. Hierbei wird eine Zeitdifferenz von 8 Stunden als kritisch angesehen. In diesem Kontext spielt es allerdings eine Rolle, ob die Unternehmen bereits langjährige positive Projekterfahrung miteinander gesammelt haben (S\_P3). Falls dies der Fall sein sollte, ist davon auszuge-



hen, dass das Problem der verschiedenen Zeitzonen bereits im Rahmen früherer Projekte diskutiert und relativiert wurde. Somit werden diese beiden SimpleRules mit einem „und“ miteinander verknüpft und ergeben somit die ComplexRule Cx1. Für den Fall, dass diese „true“ ergibt, nimmt das Programm beispielsweise „Kommunikationsrisiken aufgrund stark verschiedener Zeitzonen!“ und „Synchronisationsprobleme!“ in die Liste der potentiellen Risiken auf.

Die ComplexRules werden dabei für jedes Paar von Unternehmen die in einem Projekt zusammenarbeiten abgearbeitet. Für jedes Element der Liste von potentiellen Risiken gibt das Programm ebenfalls einen Maßnahmvorschlag zur Reduzierung bzw. Eliminierung des Risikos an.

### 7.1.3 Entity-Relationship-Diagramm des NetRiskManagers

Im Mittelpunkt des Entity-Relationship-Diagramms steht das Projekt. Es bezeichnet die Zusammenarbeit mehrerer Unternehmen. Jedes Unternehmen partizipiert durch die Projektbeteiligung an dem Projekt. Unter der Projektbeteiligung werden sowohl der Aufgabenkatalog, wie auch projektspezifische Attribute der Unternehmung zusammengefasst. Ein Unternehmen kann gleichzeitig durch mehrere Projektbeteiligungen an ebenso vielen Projekten teilnehmen. Dabei ist zu beachten, dass eine Projektbeteiligung genau zu einem Unternehmen und einem Projekt gehört. Für ein Projekt kann eine Risikomenge berechnet werden. Diese besteht aus einer Anzahl einzelner, potentieller Risiken. Diese müssen nicht projektspezifisch sein, sondern können mehreren Projekten zugeordnet werden. Diese Risiken resultieren aus ComplexRules. Die ComplexRules sind mit „und“ logisch verknüpfte SimpleRules.

Eine SimpleRule vergleicht Attribute der Unternehmung, der Projektbeteiligung und des eigentlichen Projektes untereinander oder mit Konstanten. Sie kann zu mehreren ComplexRule gehören, da, durch verschiedene Kombinationen, verschiedene Risiken auftreten können. Sollte nun eine ComplexRule erfüllt sein, so impliziert dies eine Menge von Risiken. Dabei kann ein Risiko auch durch verschiedene ComplexRule abgeleitet werden. Zu jedem Risiko existiert eine Menge von Maßnahmen, die zur Reduzierung des Risikos dienen sollen. Dabei kann durch eine Maßnahme mehrere Risiken beeinflusst werden.

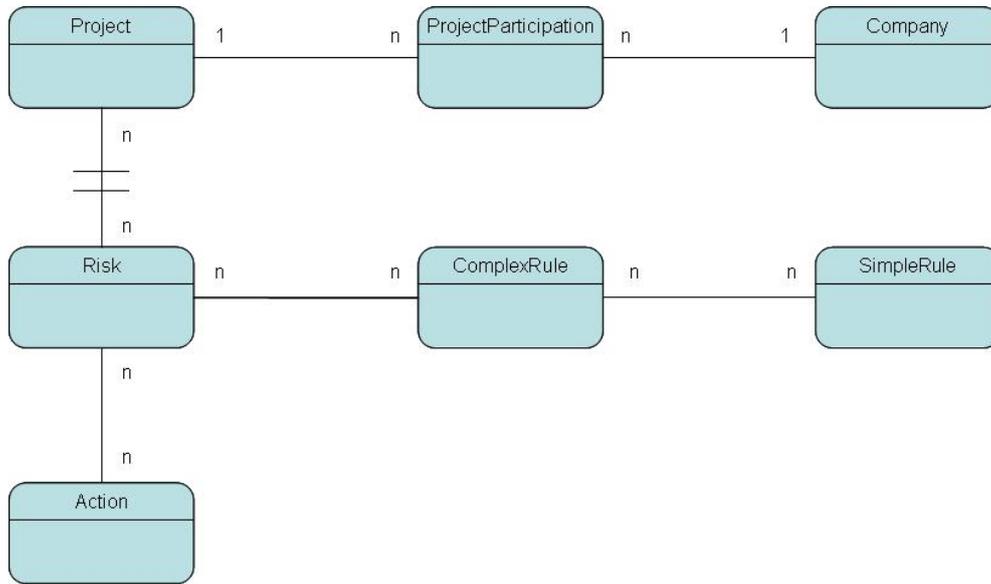
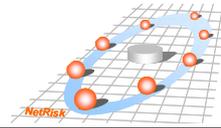


Abbildung 5-18: Entity-Relationship-Diagramm NRM

### 7.1.4 Packages des NetRiskManagers

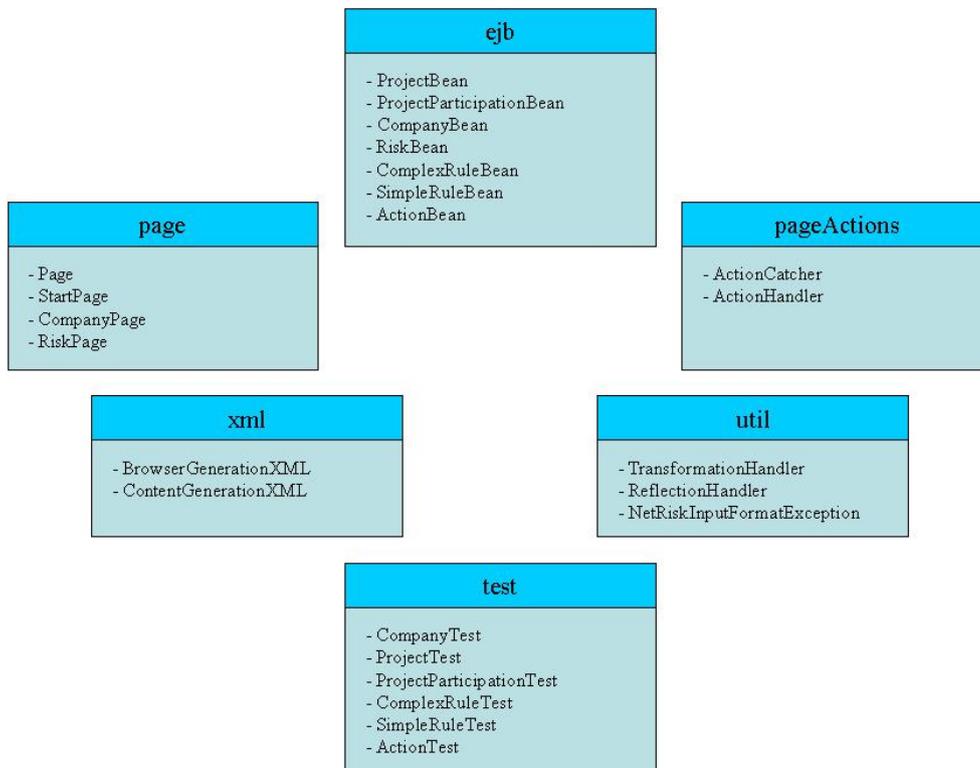


Abb. 5-19: Aufteilung in Packages - Klassen

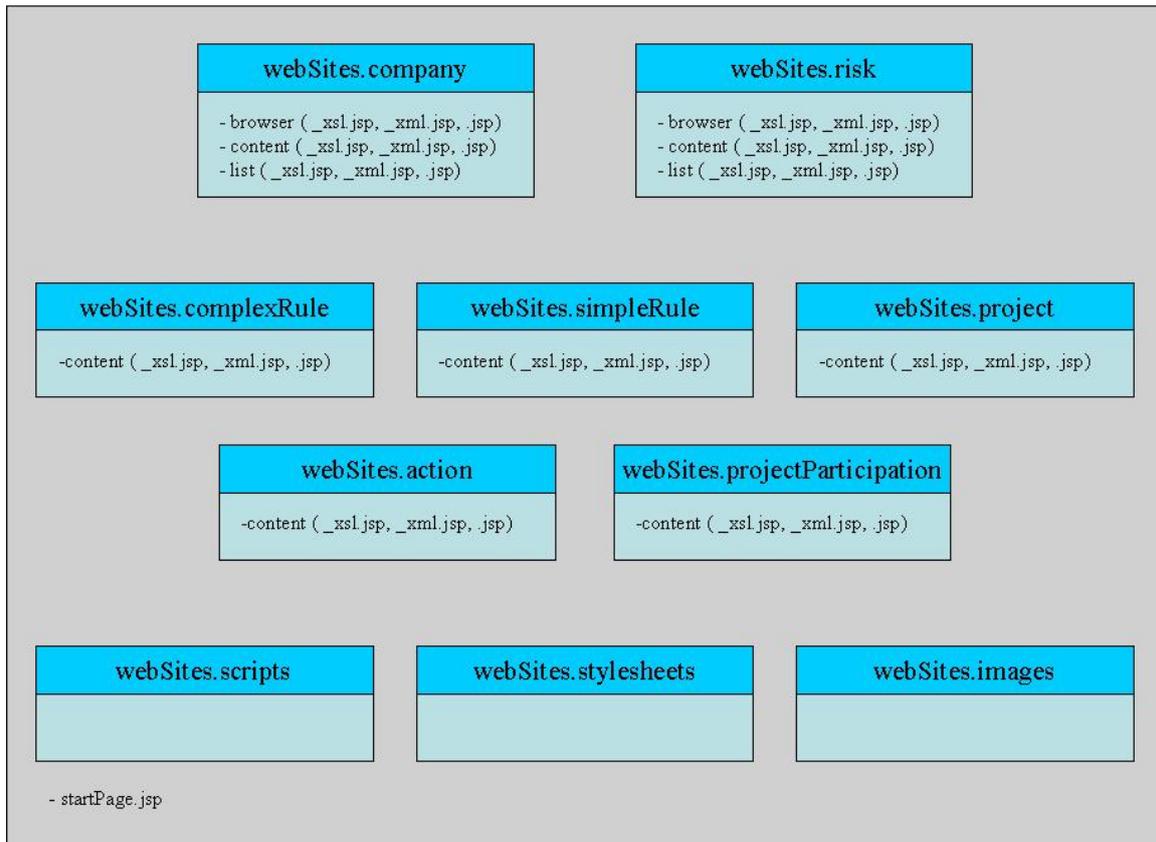
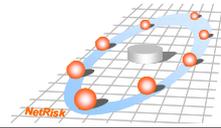
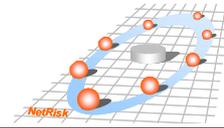


Abb. 5-20: Aufteilung in Packages – JSP



### 7.1.5 Verwendete Technologien für den NetRiskManager

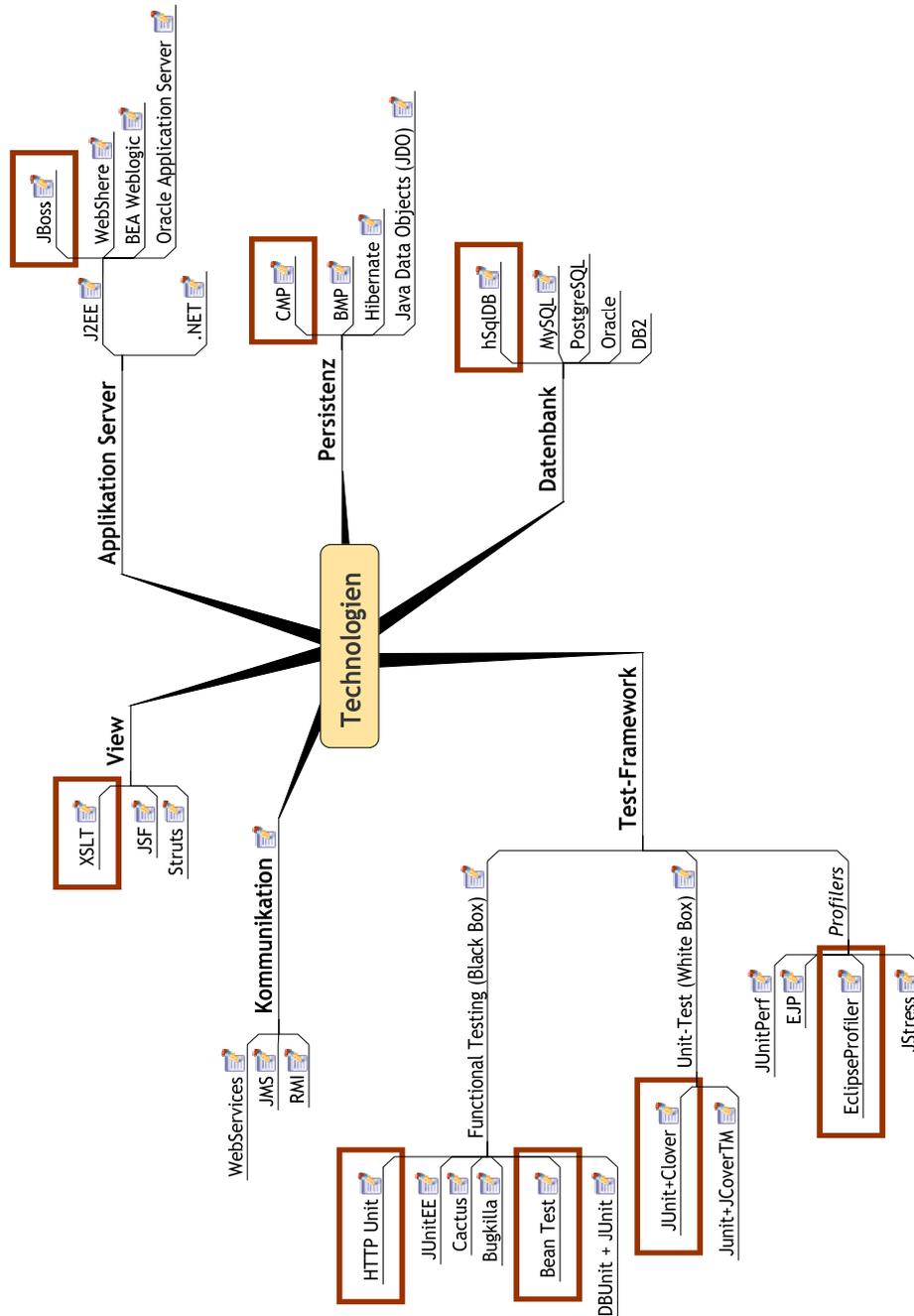


Abbildung 5-21: Technologien (benutzte Technologien umrandet)