

# Softwaretechnik C - Softwaremanagement

## LE 06: Risikomanagement



# Organisatorisches

# Agenda

- Risikomanagement
  - Einleitung
  - Risikomanagement-Prozess
  - Unsicherheiten und Risikodiagramme
  - Nutzwert vs. Risiko
  - 5 Kernrisiken einer Softwareentwicklung
  - Top Ten-Risikoliste
  - Checkliste Risikomanagement
  - Risikomanagement in Scrum

## ■ Risikomanagement

- Einleitung
- Risikomanagement-Prozess
- Unsicherheiten und Risikodiagramme
- Nutzwert vs. Risiko
- 5 Kernrisiken einer Softwareentwicklung
- Top Ten-Risikoliste
- Checkliste Risikomanagement
- Risikomanagement in Scrum



# Definition und Begriff

Risiko

Wagnis: Ereignis  
mit der  
Möglichkeit von  
Auswirkungen

Potenzielle  
Auswirkung

Positiv

Negativ

Eingetretenes  
Risiko

Chance

Problem

# Definition und Begriff

- Risikoeintritt
  - Materialisierung eines Risikos
- Eintrittsindikator
  - (Messbares) Zeichen für den Risikoeintritt
- Beispiel:
  - Risiko: Auftrag zur Entwicklung einer neuen Individualsoftware
  - Risikoeintritt: Ein Kunde entscheidet sich, einen Auftrag zu erteilen
  - Eintrittsindikator: Eingang des schriftlichen Auftrags

# Definition und Begriff

- Eintrittswahrscheinlichkeit
  - Wahrscheinlichkeit für den Eintritt eines Risikos
- Schadenshöhe
  - Quantitative Bewertung der Auswirkungen eines Risikos
  - Als Zeit oder Kosten bemessen
- Beispiel:
  - Risiko: Notebook mit Arbeitsinhalten geht verloren
  - Eintrittswahrscheinlichkeit: 10%
  - Schadenshöhe:  $2.000,- \text{ €} = 1.000,- \text{ €} \text{ (Notebook)} + 1.000,- \text{ €} \text{ (Kosten MA seit letzter Sicherung)}$

# Definition und Begriff

- Das **Ignorieren** von Risiken kann ...
  - Projekte **teurer** als geplant machen,
  - Projekte in Ihrer Laufzeit **verzögern bzw. verlängern**,
  - Projekte komplett **fehlschlagen** lassen,
  - Das Betriebsergebnis eines Unternehmens schlechter als geplant / als nötig ausfallen lassen,
  - Den Wert eines Unternehmens senken oder
  - Unternehmen gar in die **Insolvenz** treiben.

# Definition und Begriff

- Deutscher Gesetzgeber hat im April 1998 das **Gesetz zur Kontrolle und Transparenz im Unternehmensbereich (KonTraG)** beschlossen, um:
  - das **Risikobewusstsein** im Geschäftsleben zu schärfen und
  - **Vorsorge** gegen Risiken zu treffen.
- Mit diesem Gesetz wurde folgender Satz in das **Aktiengesetz** eingefügt:
  - „Der Vorstand hat geeignete Maßnahmen zu treffen, insbesondere ein Überwachungssystem einzurichten, damit den Fortbestand der Gesellschaft gefährdende Entwicklungen früh erkannt werden.“  
(AktG, § 91, Abs. 2)

# Definition und Begriff

- Das **Handelsgesetzbuch** wurde dabei wie folgt ergänzt:
  - „Im Lagebericht ist auch auf die Risiken der künftigen Entwicklung einzugehen“ (HGB, § 289, Absatz 1 und § 315, Absatz 1)“
- Somit kann bereits **fehlende Vorsorge Schadensersatzforderungen** nach sich ziehen!
- Vorstand und Aufsichtsrat müssen hierbei beweisen, dass ein Organisationsverschulden ihrerseits nicht vorliegt
  - Umkehr der Beweislast (AktG, § 93, Absatz 2)
  - Es muss (seitens der Gesellschaft) der Nachweis geführt werden, dass ein funktionsfähiges Risikofrühherkennungssystem etabliert ist, das vom Abschlussprüfer geprüft wird.

# Definition und Begriff

- Das Institut der Wirtschaftsprüfer (IDW) hat folgende **Mindestanforderungen an ein Risikofrühkennungssystem** aufgestellt:
  1. Sämtliche **betriebliche Prozess- und Funktionsbereiche** sind zu integrieren
  2. Definition von **Risikofeldern**
  3. Wirksame **Risikoerkennung und Risikoerfassung**
  4. **Risikoanalyse** und Beobachtung der Risikoveränderung
  5. Beurteilung der Tragweite der erkannten Risiken in Bezug auf **Eintrittswahrscheinlichkeit** und **quantitative Auswirkung**
  6. Flexibilität der **Risikokommunikation** und **Risikoberichterstattung**
  7. Eindeutige Zuordnung von **Verantwortlichkeiten** und **Aufgaben**
  8. Dokumentation des **Risikomanagement-Prozesses** in Form eines Risikohandbuchs

# Beispiele für Risiken

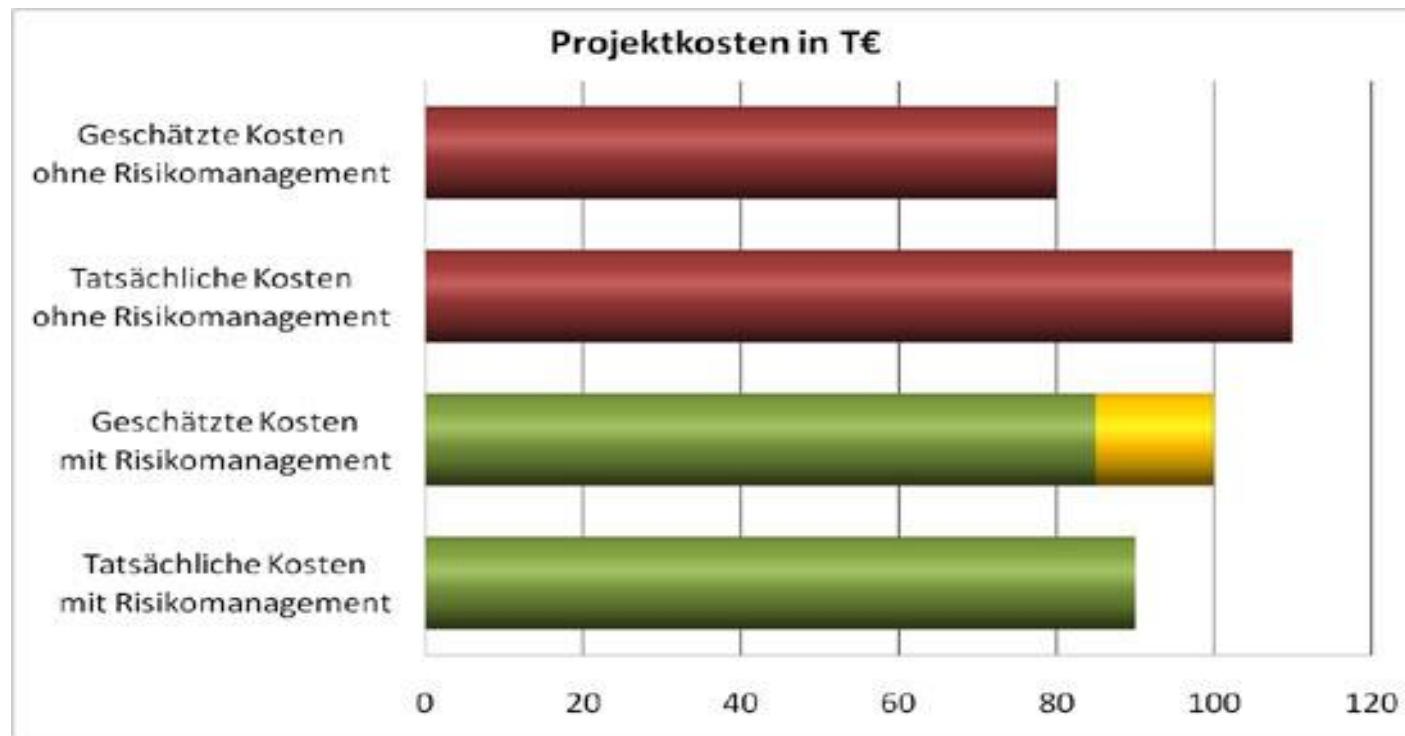
Risiko	Art des Risikos	Beschreibung
Personalveränderung	Projekt	Erfahrenes Personal verlässt das Projekt vor seiner Fertigstellung.
Managementwechsel	Projekt	Es gibt einen Wechsel zu einem neuen Management, das neue Prioritäten setzt.
Nichtverfügbarkeit von Hardware	Projekt	Für das Projekt unverzichtbare Hardware wird nicht pünktlich geliefert.
Veränderung der Anforderungen	Projekt und Produkt	Die Anzahl der Änderungen bei den Anforderungen ist größer als erwartet.
Verzögerungen in der Spezifikation	Projekt und Produkt	Die Spezifikationen wichtiger Schnittstellen sind nicht pünktlich verfügbar.
Unterschätzen des Umfangs	Projekt und Produkt	Der Umfang des Projekts wurde unterschätzt.
Unzureichende Leistung der CASE-Werkzeuge	Produkt	CASE-Werkzeuge, die das Projekt unterstützen, bringen nicht die erwartete Leistung.
Technologie-veränderung	Geschäft	Die dem System zugrunde liegende Technologie wird durch eine neue Technologie verdrängt.
Produktkonkurrenz	Geschäft	Ein Konkurrenzprodukt wird auf den Markt gebracht, bevor das System fertiggestellt ist.

# Beispiele für Risikofaktoren

Art des Risikos	Mögliche Risiken
Technologisch	<p>Die im System verwendete Datenbank kann nicht so viele Transaktionen pro Sekunde durchführen wie erwartet. (1)</p> <p>Zur Wiederverwendung vorgesehene Softwarekomponenten enthalten Fehler, die ihre Funktionalität einschränken. (2)</p>
Personenbezogen	<p>Es ist unmöglich, genügend Personal mit den benötigten Fähigkeiten zu rekrutieren. (3)</p> <p>Wichtige Mitarbeiter sind krank oder zu kritischen Zeitpunkten nicht verfügbar. (4)</p> <p>Es gibt keine Möglichkeit, erforderliche Schulungen durchzuführen. (5)</p>
Unternehmensbezogen	<p>Das Unternehmen wird umstrukturiert, sodass ein anderes Management für das Projekt verantwortlich ist. (6)</p> <p>Finanzielle Probleme des Unternehmens zwingen zu Kürzungen des Projekt-budgets. (7)</p>
Werkzeuge	<p>Der durch entsprechende Softwarewerkzeuge generierte Code ist ineffizient. (8)</p> <p>Softwarewerkzeuge arbeiten nicht wie erwartet durchgängig zusammen. (9)</p>
Anforderungen	<p>Es werden Änderungen der Anforderungen vorgeschlagen, die eine beträchtliche Nachbearbeitung des Entwurfs nach sich ziehen. (10)</p> <p>Kunden verstehen die Auswirkungen von Anforderungsänderungen nicht. (11)</p>
Schätzung	<p>Die zur Entwicklung der Software benötigte Zeit wird unterschätzt. (12)</p> <p>Die Anzahl der Fehlerbehebungen wird unterschätzt. (13)</p> <p>Der Umfang der Software wird unterschätzt. (14)</p>

# Risikomanagement

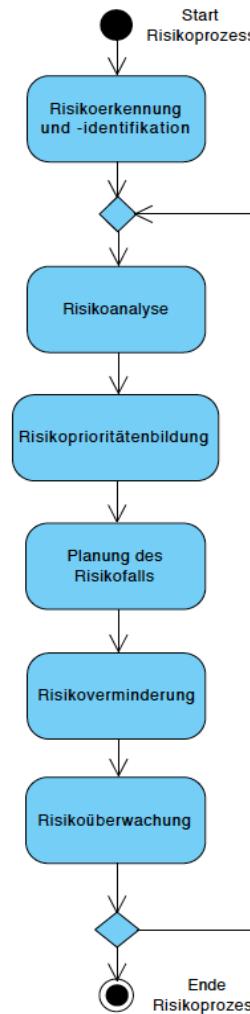
- Unter **Risikomanagement** wird der **planvolle und strukturierte Umgang mit Risiken** verstanden
- Durch Risikomanagement sollen mögliche Risiken **antizipiert** werden



## ■ Risikomanagement

- Einleitung
- Risikomanagement-Prozess
- Unsicherheiten und Risikodiagramme
- Nutzwert vs. Risiko
- 5 Kernrisiken einer Softwareentwicklung
- Top Ten-Risikoliste
- Checkliste Risikomanagement
- Risikomanagement in Scrum

# Risikomanagement-Prozess



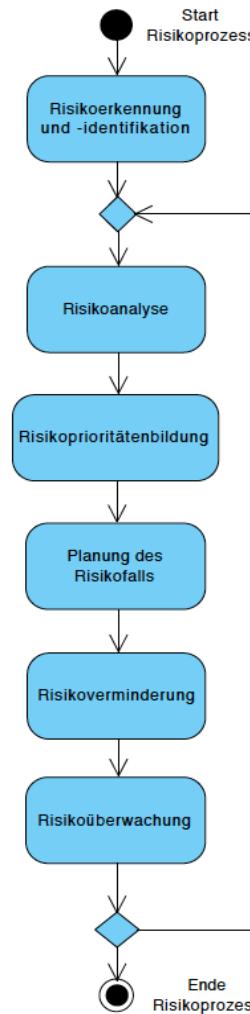
- Um Projektrisiken zu managen, ist es nützlich, im Projekt einen eigenen **Risikomanagement-Prozess** zu etablieren

- 1. Schritt: Risikoidentifikation
- 2. Schritt: Risikoanalyse
- 3. Schritt: Risikoprioritätenbildung
- 4. Schritt: Planung des Risikofalls
- 5. Schritt: Risikoverminderung
- 6. Schritt: Risikoüberwachung

Schritte der  
**Risikobewertung**

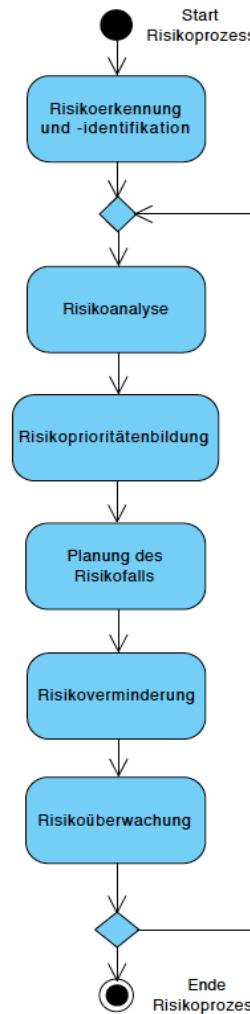
Schritte der  
**Risikobeherrschung**

## 1. Schritt: Risikoidentifikation



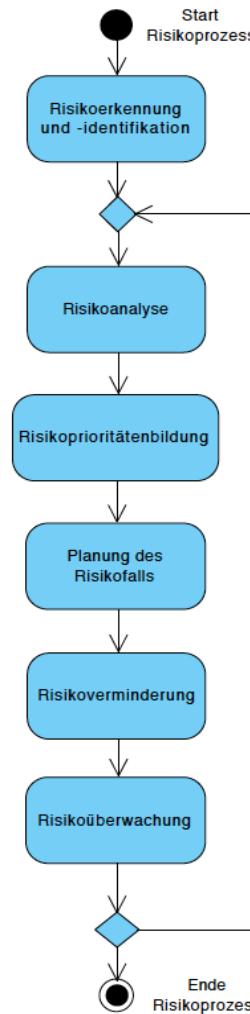
- Mithilfe von Brainstorming oder Checklisten werden mögliche Risiken **identifiziert**
- **Ergebnis:** Liste der **projektspezifischen Risikoelemente**, die Projekterfolg **potenziell gefährden** bzw. negativ beeinflussen könnten
- Zur genaueren Einschätzung eines Risikos können **Risiko-Wahrscheinlichkeitstabellen** genutzt werden
- **Post-Mortem-Analyse** vergangener Projekte (erfolgreich und nicht erfolgreich)
- **Abweichungen und deren Ursache** identifizieren
  - Jede Ursache = ein potenzielles Projektrisiko

## 2. Schritt: Risikoanalyse



- Schätzung von Eintrittswahrscheinlichkeit und Schadenshöhe je Risikoelement
- Für eine **quantitative Bewertung** eines Risikos wird der **Risikofaktor** berechnet
- **Risikofaktor** = Eintrittswahrscheinlichkeit \* Schadenshöhe
- Risikofaktor kann auch in erwarteten **Verzögerungsmonaten** oder **Mehrkosten** für das Projekt ausgedrückt werden
- Nicht alle Risiken müssen verwaltet werden:
  - Risiko kann ignoriert werden, wenn Eintrittswahrscheinlichkeit  $\approx 0$
  - Risiko kann ignoriert werden, wenn Schadenshöhe  $\approx 0$

## 3. Schritt: Risikoprioritätenbildung



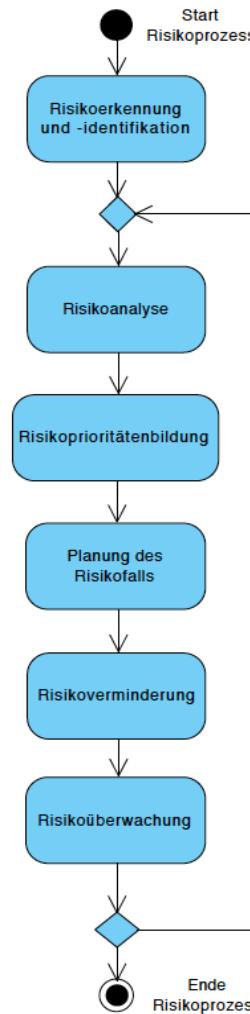
- **Ergebnis der Risikoidentifikation** können **zahlreiche Risikoelemente** sein
- Ziel der **Risikoprioritätenbildung**: es soll verhindert werden, dass die wirklich relevanten Risiken **übersehen** werden
  - Priorisierung gemäß Höhe der Risikofaktoren
- Risikofaktoren werden oftmals in einer **Risikomatrix** dargestellt
  - x-Achse: Schadenshöhe
  - y-Achse: Eintrittswahrscheinlichkeit
- Anhand der Risikomatrix lassen sich die relevanten Risiken, die wesentlichen Risiken und die berichtspflichtigen Risiken herausfiltern

## 3. Schritt: Risikoprioritätenbildung

		Eintrittswahrscheinlichkeit						
		5/A	5/B	5/C	5/D	5/E		
5 sehr wahrscheinlich (75 % <= X <= 100 %)	4/A	4/B	4/C	4/D	4/E			
	3/A	3/B	3/C	3/D	3/E			
4 wahrscheinlich (50 % <= X < 75 %)	2/A	2/B	2/C	2/D	2/E			
	1/A	1/B	1/C	1/D	1/E			
		Schadenshöhe						
		A unbedeutend (0 TEuro < X <= 25 TEuro)	B gering (25 TEuro < X <= 50 TEuro)	C mittel (50 TEuro < X <= 100 TEuro)	D schwerwiegend (100 TEuro < X <= 1 Mio. Euro)	E existenzbedrohend (1Mio. Euro < X)		

Handlungsbedarf im Rahmen des Risikomanagements	Risikoverfolgung
unbedingt Handlungsbedarf (■ ad hoc Bericht an Vorstand und Risikomanager)	Verfolgung im Rahmen des Risikomanagements
Handlungsbedarf (■ ad hoc Bericht an Vorstand und Risikomanager)	
unter Umständen Handlungsbedarf	
kein Handlungsbedarf	Verfolgung durch Bereichsleiter
kein relevantes Risiko	

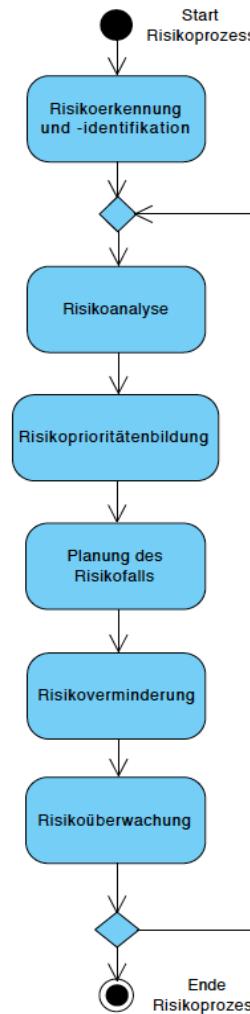
## 4. Schritt: Planung des Risikofalls



- **Nach Eintritt eines Risikofalls**
- Für die **kritischen Risiken** müssen sogenannte **Eventualfallpläne** erstellt werden
  - In diesen wird festgelegt, was zu tun ist, wenn ein kritisches Risiko (also mit entsprechend hohem Risikofaktor) eintritt
  - Zur Risikobegrenzung bzw. -eindämmung wird (bereits bei der Projektplanung) eine Zeit- und/oder Geldreserve zurückgestellt
  - Die Rückstellung korreliert mit dem entsprechenden Risikofaktor

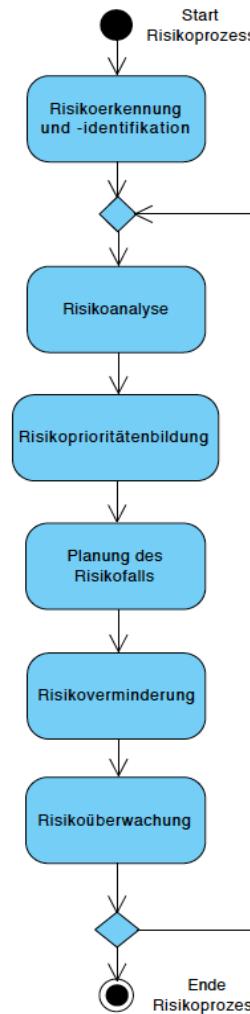
# Risikomanagement-Prozess

## 5. Schritt: Risikovermeidung



- **Vor Eintritt** eines Risikofalls
- Es wird festgelegt, **welche Maßnahmen** vor einem eventuellen Eintritt eines Risikos erfolgen
- Ziel: **geplante Maßnahmen** zur Risikobewältigung im Schadensfall **möglich und effektiv** machen
- Es wird also so getan, als ob das Risiko in jedem Fall eintritt
- Die **Kosten und durchzuführenden Maßnahmen** zur **Risikovermeidung** entstehen somit auch, wenn das Risiko **nicht eintritt**

## 6. Schritt: Risikoüberwachung

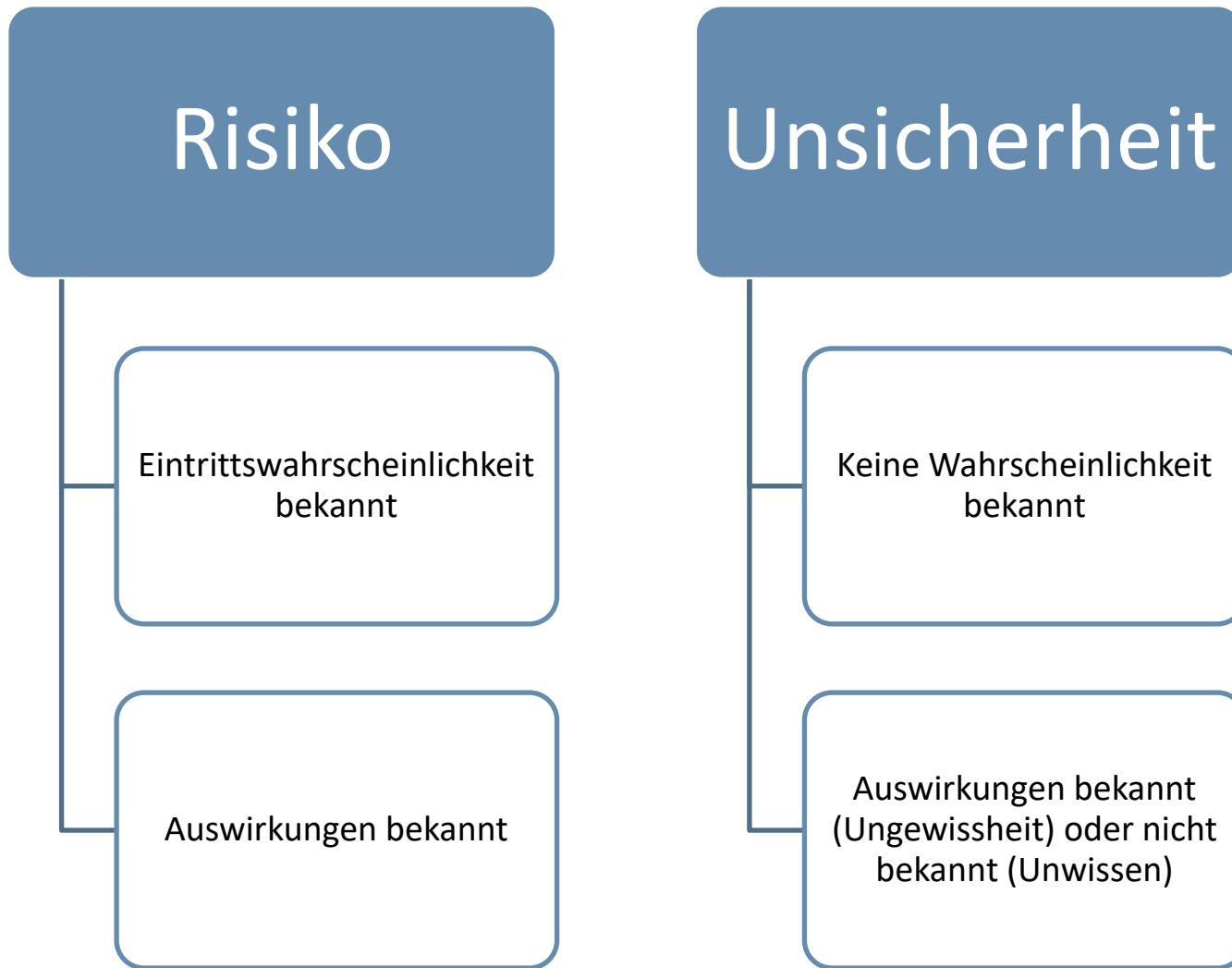


- **Kontinuierliche Überwachung** der **Eintrittsindikatoren**, die angeben, ob ein Risiko eingetreten ist
- Gibt es zu einem Risiko mehrere Eintrittsindikatoren, dann ist zu beachten, dass der früheste Indikator möglicherweise auch ein falsches positives Signal liefern kann

## ■ Risikomanagement

- Einleitung
- Risikomanagement-Prozess
- Unsicherheiten und Risikodiagramme
- Nutzwert vs. Risiko
- 5 Kernrisiken einer Softwareentwicklung
- Top Ten-Risikoliste
- Checkliste Risikomanagement
- Risikomanagement in Scrum

# Risiko vs. Unsicherheit



# 10 Hauptquellen der Unsicherheit in der SWE

1. Anforderungen
  - Was soll das SW-Produkt leisten?
2. Zusammenspiel
  - Wie soll das SW-Produkt mit den Benutzern und anderen SW-Produkten und –Systemen zusammenspielen?
3. Umgebungsveränderungen
  - Wie ändern sich die Bedürfnisse und Ziele während des Softwareentwicklungsprozesses?
4. Ressourcen
  - Welche Mitarbeiter mit welchen Fähigkeiten stehen im Projekt zur Verfügung?
5. Management
  - Ist das Management in der Lage, produktive Teams zusammenzustellen, die Mitarbeiterfluktuation zu minimieren und komplexe Teilprojekte zu koordinieren?

# 10 Hauptquellen der Unsicherheit in der SWE

## 6. Supply Chain

- Erbringen andere Projektbeteiligte die erwartete Leistung?

## 7. Politik

- Wie wirken sich politische Machtspiele aus, die unrealistische Ziele und Vorgaben machen, und dadurch den Projekterfolg gefährden?

## 8. Interessen / Konflikte

- Wie werden unterschiedliche Interessen der Projektbeteiligten berücksichtigt?

## 9. Innovation

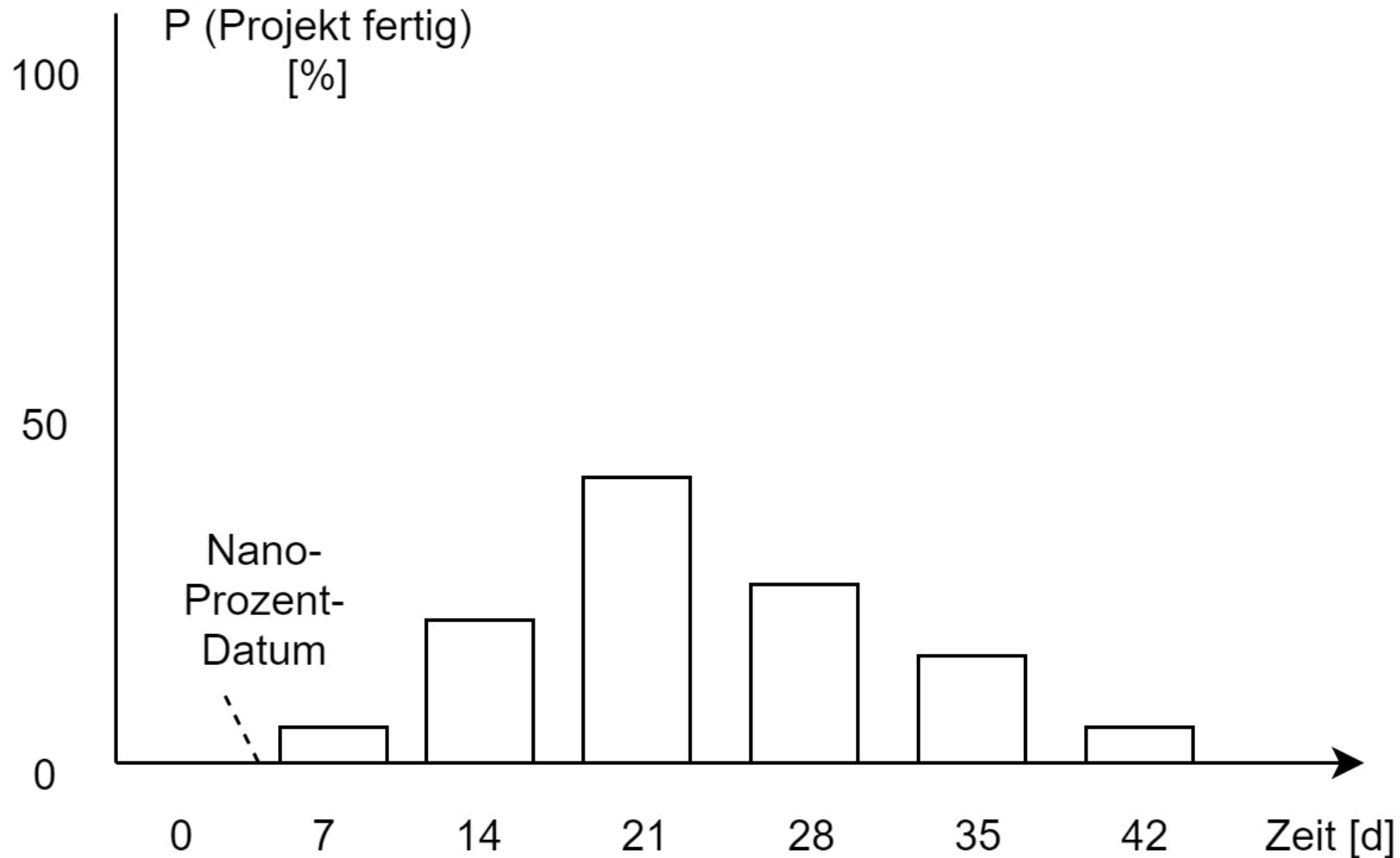
- Wie beeinflussen neue Technologien und Methoden den Projekterfolg?

## 10. Skalierung

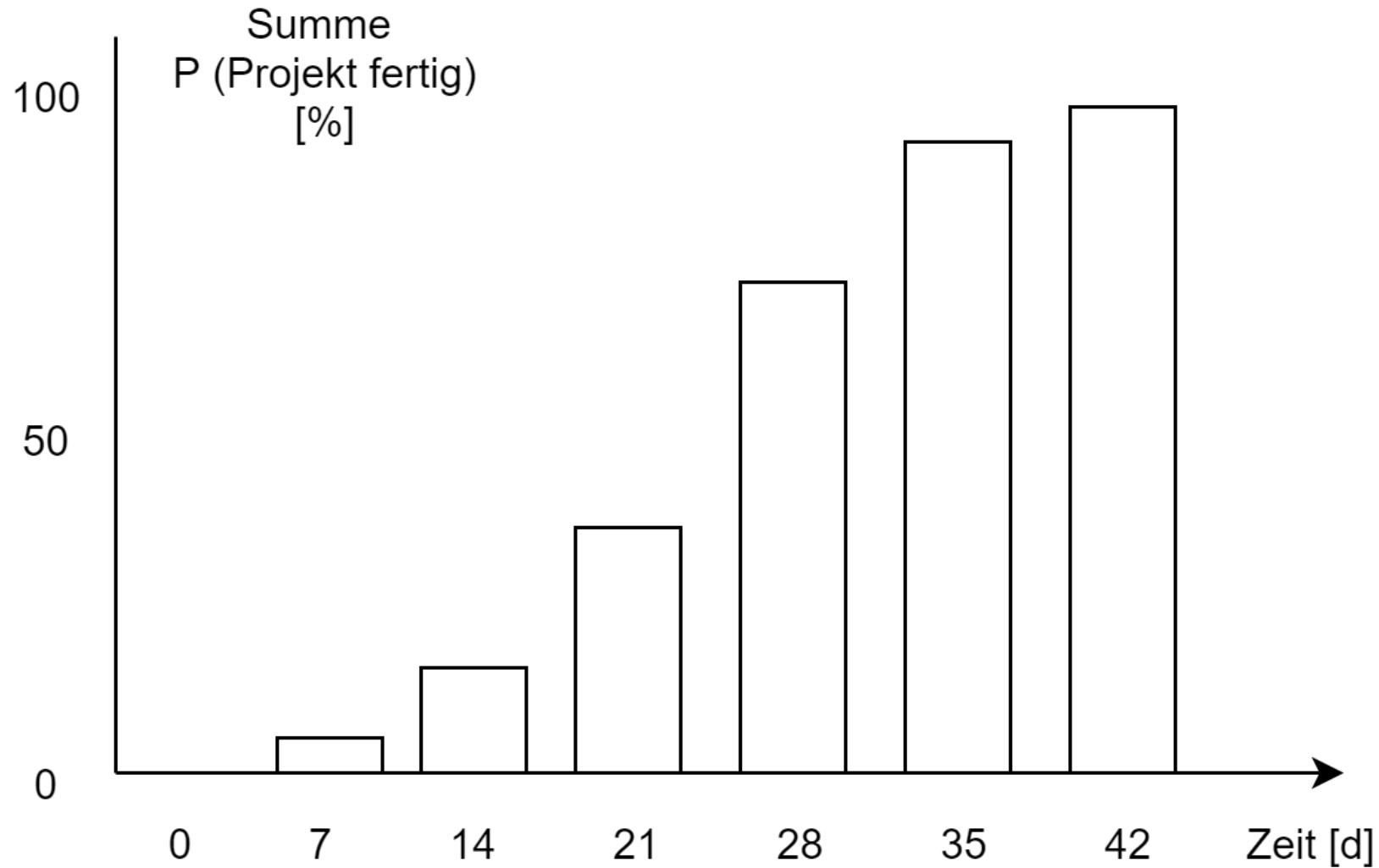
- Wie wirken sich zusätzliche Funktionen auf die Projektleistung aus?

# Unsicherheiten in der SWE

## Visualisierung durch Unsicherheitsdiagramm



# Unsicherheitsdiagramm

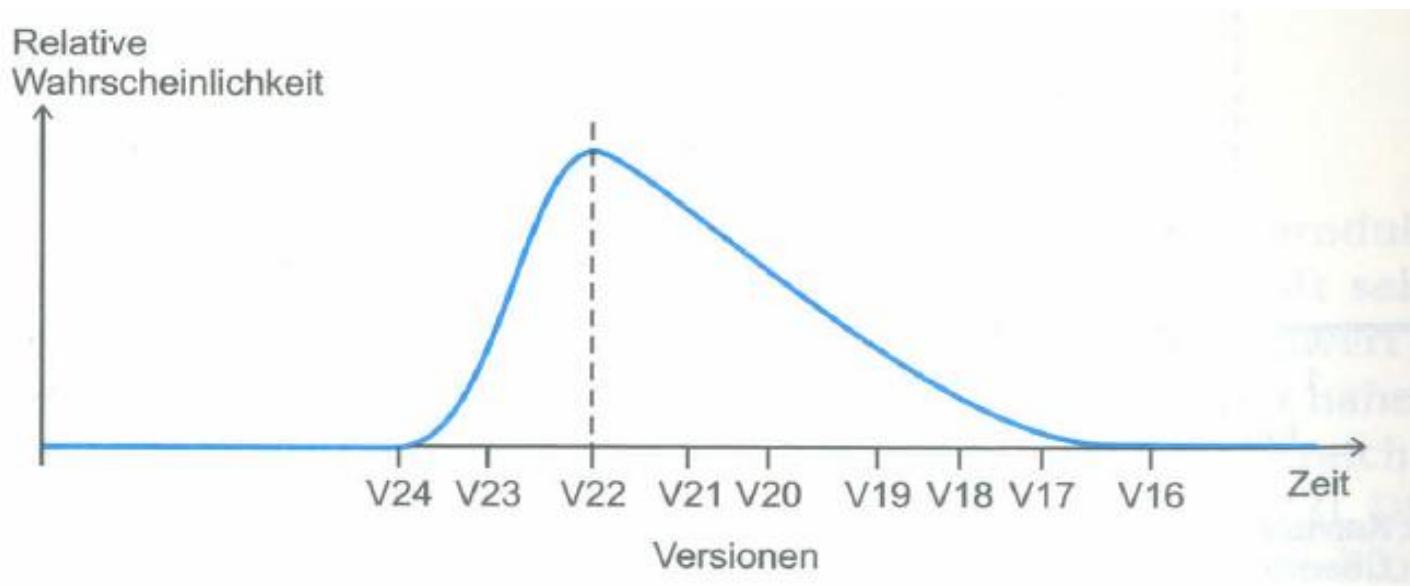


# Unsicherheiten in der Softwareentwicklung

- Die Abzweigung des Graphen von der x-Achse definiert **das erste Datum**, dessen Wahrscheinlichkeit nicht bei Null, aber doch im **Nanoprozent**-Bereich liegt
  - Wird daher auch als **Nano-Prozent-Datum N** bezeichnet
  - In der Regel wird dieser Termin bei der **Aufwandsschätzung** ermittelt
  - **Fehler** liegt darin, diesen Termin als **Fertigstellungstermin** festzulegen
  - Realistisch liegt die Fenstergröße in der SWE in einem Bereich von **150 bis 200 Prozent von N**
    - => bei einem Projekt, bei dem  $N = 6$  Monate, endet der Unsicherheitsgraph ggf. erst 12 Monate nach N bzw. 18 Monate nach Projektstart!

# Unsicherheiten in der Softwareentwicklung

- Tipp: Geben Sie Ihrem Projektteam ein gewagtes Ziel vor, das irgendwo zwischen N und dem Termin liegt, der dem Auftraggeber zugesagt wurde
- Manche **Termine können nicht frei gewählt** werden (Jahreswechsel, Messen, Wahlen, etc.)
- **Lösung:** Terminunsicherheit gegen **Funktionsunsicherheit** tauschen



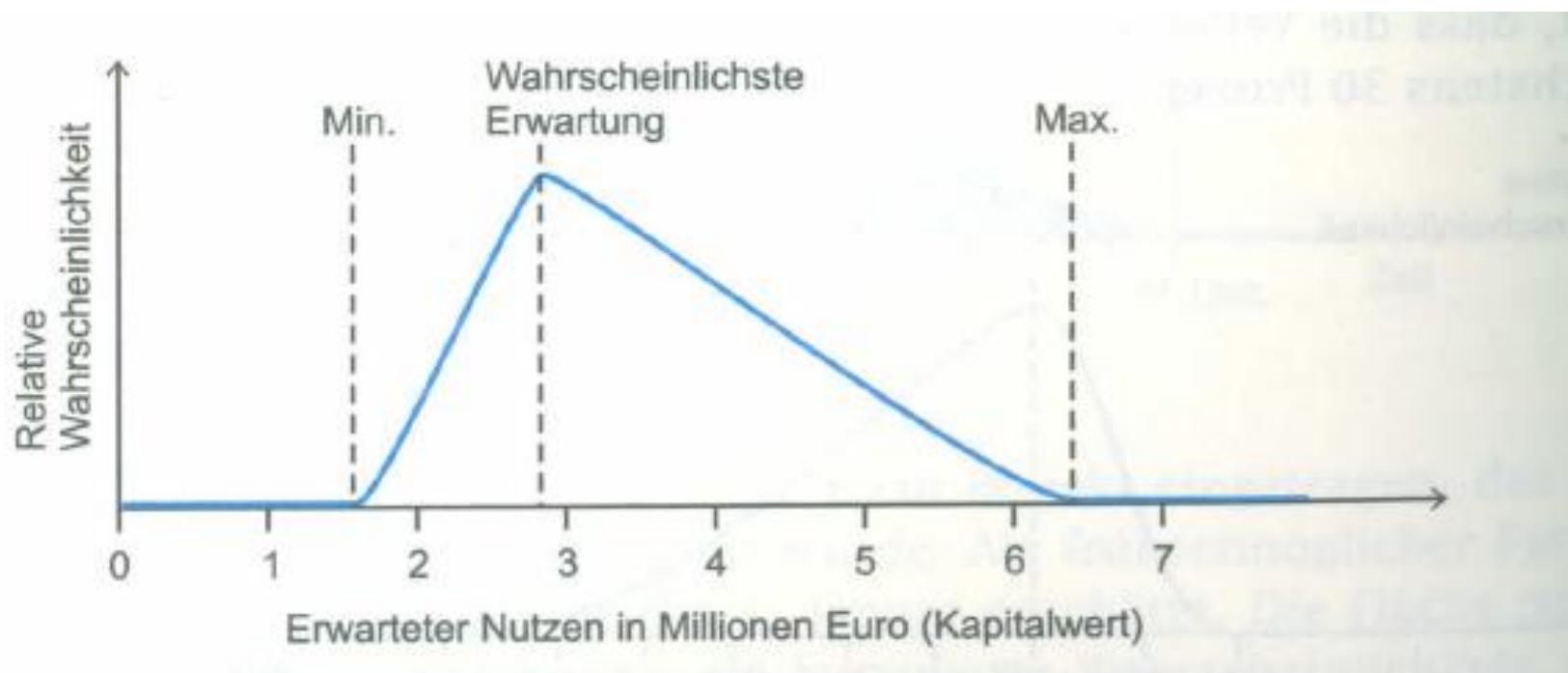
## ■ Risikomanagement

- Einleitung
- Risikomanagement-Prozess
- Unsicherheiten und Risikodiagramme
- Nutzwert vs. Risiko
- 5 Kernrisiken einer Softwareentwicklung
- Top Ten-Risikoliste
- Checkliste Risikomanagement
- Risikomanagement in Scrum

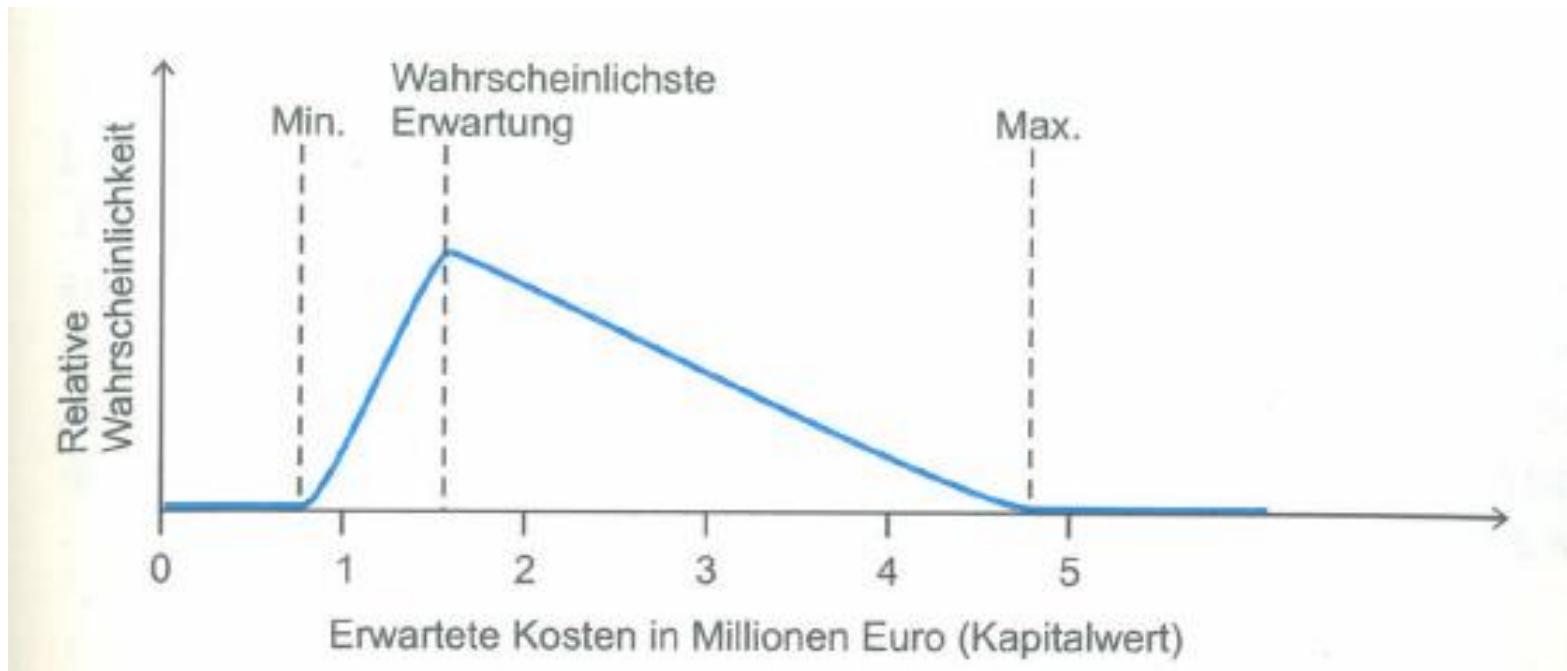
# Nutzwert vs. Risiko

- Hohe **Risiken** werden oftmals nur bei hohen **Chancen** eingegangen
- Um **Nutzen und Risiken vergleichen** zu können, ist es nötig, den **Nutzen eines Softwareprodukts zu quantifizieren**
  - Schwierig, denn auch der Nutzwert ist mit Unsicherheiten behaftet
- **Unsicherheit des Nutzens** kann wie folgt abgeschätzt werden:
  - Die optimistische und die pessimistische Einschätzung werden als Umsatz oder Gewinn in Euro quantifiziert
  - Zwischen beiden Polen liegt der wahrscheinlichste Nutzen
  - Diese drei Werte ergeben ein erstes Unsicherheitsdiagramm, das die Risiken eingrenzt, die mit dem Nutzwert verbunden sind

# Unsicherheitsdiagramm des Projektnutzens

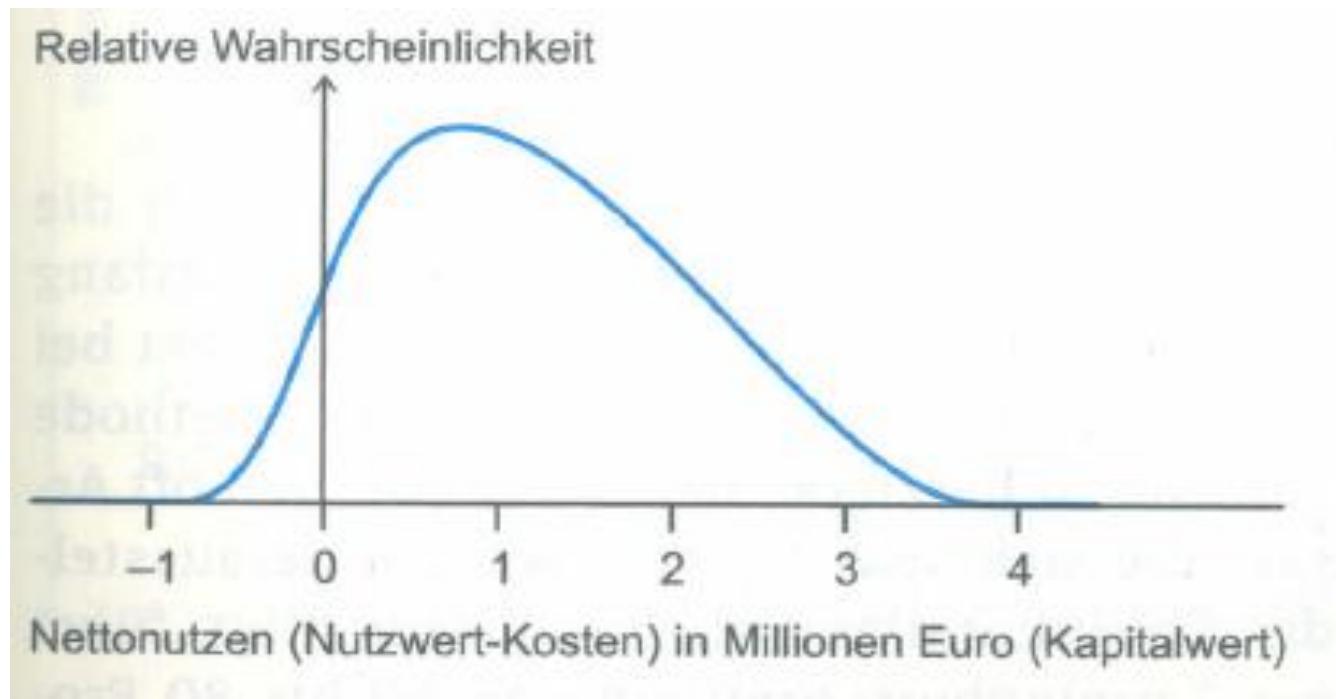


# Unsicherheitsdiagramm des Projektrisikos



## Nettonutzendiagramm

- **Vergleich von Nutzen und Risiko => Nettonutzen**
- Ist durch eine Risikosimulation möglich (100 simulierte Projekte), die ein Nettonutzendiagramm ergibt
  - 1 Projekt von 10 Projekten erbringt KEINEN Nettonutzen
  - In 9 von 10 Fällen ist ein Nettonutzen zu erwarten



# Nutzwert

- **Nutzwert** ist oftmals **sehr ungleichmäßig** über das Softwareprodukt verteilt
- Der **wahre Geldwert eines Softwareprodukts** liegt in bestimmten **Schlüsselfunktionen** im oder nahe beim Produktkern
- Manchmal umfasst dieser Kernbereich nicht mehr als 10% des Quellcodes
- => neben den geschätzten Kosten sollte jedem Teilprodukt auch sein erwarteter Nutzen zugeordnet werden
- Aus diesen Angaben kann dann das Nutzwert-Kosten-Verhältnis für jeden Systemteil berechnet werden (vgl. Kano-Modell)
- => Teilprodukte, die ein hohes Nutzwert-Kosten-Verhältnis aufweisen, sollten zuerst fertiggestellt werden

## ■ Risikomanagement

- Einleitung
- Risikomanagement-Prozess
- Unsicherheiten und Risikodiagramme
- Nutzwert vs. Risiko
- 5 Kernrisiken einer Softwareentwicklung
- Top Ten-Risikoliste
- Checkliste Risikomanagement
- Risikomanagement in Scrum

# 5 Kernrisiken einer Softwareentwicklung

1. Fehlerhafter Zeitplan
2. Inflation der Anforderungen
3. Mitarbeiterfluktuation
4. Spezifikationskollaps
5. Geringe Produktivität

- Siehe auch:
  - Balzert, 2008, S. 370 ff.
  - DeMarco, Lister, 2003, S. 108 ff.

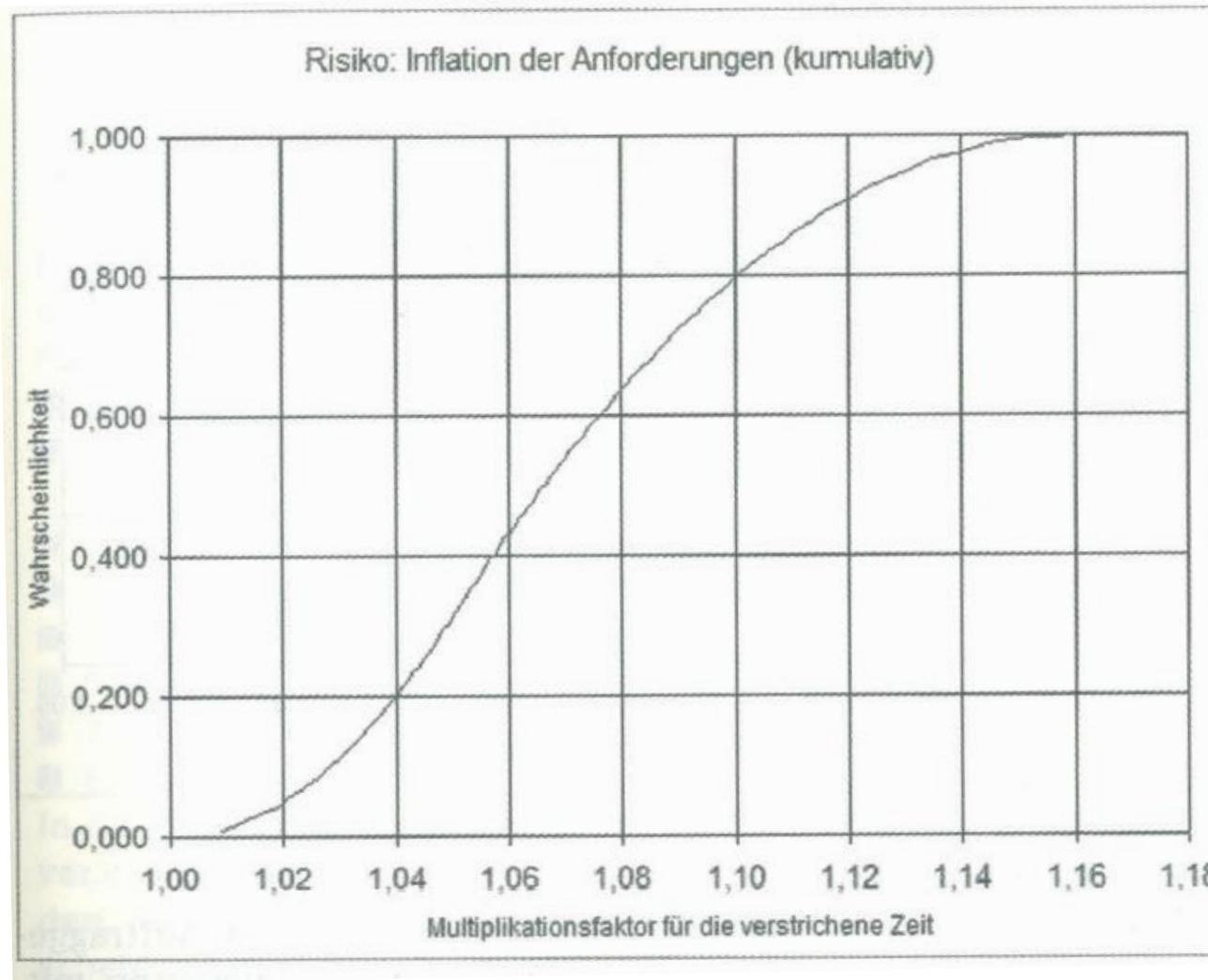
# 1. Fehlerhafter Zeitplan

- **Fehlerhafter Zeitplan** verursacht von allen fünf Kernrisiken die **größte Soll-Ist-Abweichung**
- **Ursache: falsche Aufwandsschätzungen** trotz guter Methoden (Cocomo, Function Points, PERT-3-Punkt, Schätzklausur etc.)
- Problem: notwendige Aufgaben werden **übersehen**
- Oft wird eine detaillierte Aufwandsschätzung vernachlässigt und der **Zeitplan aus der Terminvorgabe abgeleitet**
- Bei kleineren Projekten sind die Terminüberschreitungen im relativen Vergleich größer als bei Großprojekten
- Beispiel:
  - Um den Toll-Collect-Auftrag zu erhalten, veranschlagte die beauftragte Firma eine Projektdauer von weniger als einem Jahr. Diese Projektdauer hielt sogar der Auftraggeber nachträglich für unrealistisch.

## 2. Inflation der Anforderungen

- Auftraggeber entdeckt **während der Softwareentwicklung neue Anforderungen** oder modifiziert bereits vorhandene Anforderungen
  - Das ist in der Praxis eher die Regel als die Ausnahme
- => unbedingt von vornherein einen Puffer für neu hinzukommende bzw. veränderte Anforderungen einplanen
  - De Marco und Lister (2003, S. 114) haben dazu Daten von Softwareentwicklungsprojekten, die ein bis zwei Jahre dauerten und bis zu zehn Mitarbeiter(innen) umfassten, ausgewertet.
  - Das Ergebnis zeigt die folgende Abbildung

## 2. Inflation der Anforderungen



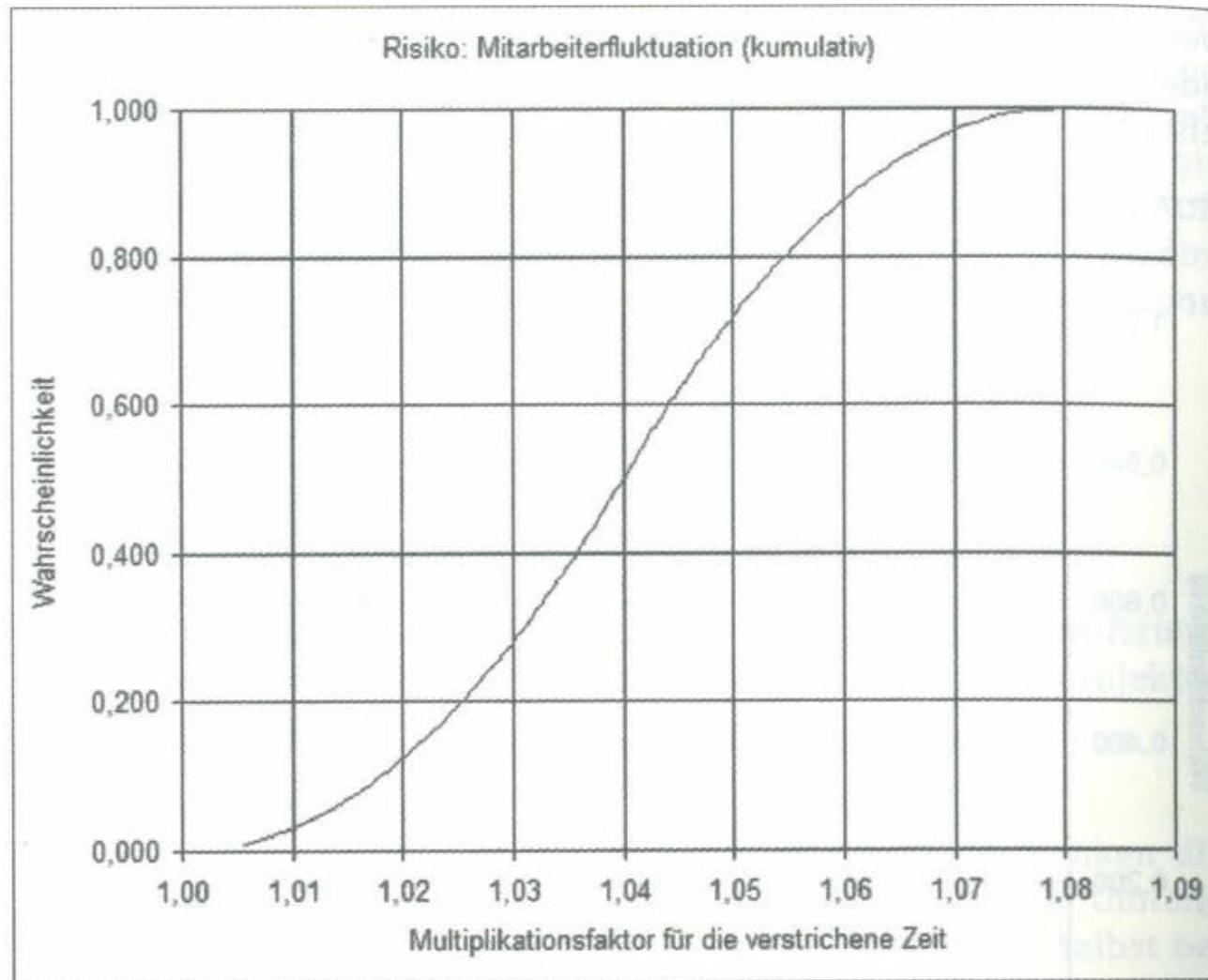
## 2. Inflation der Anforderungen

- Liegt eine Wahrscheinlichkeit von 80% für neue Anforderungen vor, dann ist mit einem Zeitverzögerungsfaktor von 1,10 zurechnen
- Ein Projekt, das ursprünglich 10 Monate dauern sollte, benötigt dann also 11 Monate
- Somit kann und sollte direkt von Projektbeginn an ein einmonatiger zeitlicher Puffer zusätzlich eingeplant werden!

### 3. Mitarbeiterfluktuation

- Natürlicher Effekt in der Berufs- und Arbeitswelt
- Analog zum Kernrisiko Inflation der Anforderungen sollte ein Puffer für die Mitarbeiterfluktuation eingeplant werden
- Zusätzlich zu berücksichtigen ist die Einarbeitungszeit für eine Ersatzperson
- Zur Verminderung dieses Risikos sollte:
  - zu Projektbeginn eine Vertreterregelung eingeführt werden
  - die Dokumentation entsprechend nachvollziehbar ausgelegt sein
- Wie sich die Mitarbeiterfluktuation auf Ein- und Zwei-Jahres-Projekte auswirkt, zeigt die folgende Abbildung, wobei eine branchenübliche Fluktuationsrate zu Grunde gelegt wurde

### 3. Mitarbeiterfluktuation



## 4. Spezifikationskollaps

- Bei der Definition der Anforderungen gibt es oftmals unterschiedliche Interessen auf Auftraggeberseite
  - Problem, dass oftmals (nicht ausgetragene) Konflikte in unklaren Anforderungen versteckt sind
  - Erkenntnis: Es ist möglich, ein Softwareprodukt unklar zu spezifizieren; es ist aber nicht möglich, ein Softwareprodukt unklar zu implementieren
- SWE-Projekt kann auch dadurch boykottiert werden, dass am Ende der Spezifikationsphase überflüssige Anforderungen ergänzt werden, die das Ende der Spezifikationsphase verhindern
- 10% bis 15% aller Projekte werden vor Fertigstellung eingestellt:
  - DeMarco und Lister (2003, S. 118) schlagen daher vor, für jedes Projekt das Risiko „Projekteinstellung“ vorzusehen
  - Erst nach abgeschlossener Spezifikation ist dieses Risiko gleich Null

## 5. Geringe Produktivität

- Die Leistungen von Mitarbeiter(inne)n sind innerhalb beliebiger Teams sehr unterschiedlich
- Unterschiede zwischen dem/der leistungsstärksten und dem leistungsschwächsten Mitarbeiter(in) sind immer größer als die Leistungsunterschiede verschiedener Teams
- Außer bei sehr kleinen Teams oder bei Einzelpersonen ist die Wahrscheinlichkeit, dass die Leistung ein Projekt positiv beeinflusst, genauso groß, wie dass sie es negativ beeinflusst

## ■ Risikomanagement

- Einleitung
- Risikomanagement-Prozess
- Unsicherheiten und Risikodiagramme
- Nutzwert vs. Risiko
- 5 Kernrisiken einer Softwareentwicklung
- **Top Ten-Risikoliste**
- Checkliste Risikomanagement
- Risikomanagement in Scrum

# Top Ten Risikoliste

- Bewährte **Technik zur Risikoüberwachung**
- Ermöglicht es Risikomanager:innen, sich auf die hohen Risiken und kritischen Erfolgsfaktoren zu konzentrieren
- Keine Überlastung mit Details, die nur geringe Risiken enthalten
- **Verfolgung der Top Ten-Risiken** umfasst folgende Schritte:
  1. Risikoelemente in eine Rangfolge bringen
  2. Festlegung regelmäßiger Überprüfungstermine durch das höhere Management
  3. Jede Sitzung beginnt mit einem Bericht über den Fortschritt bei den Top Ten-Risikoelementen (Rangordnung, Rang bei der letzten Sitzung, Häufigkeit der Nennung, Entwicklung seit der letzten Sitzung)
  4. In der Sitzung sollte es darum gehen, die Risikoelemente zu beseitigen

# Top Ten Risikoliste

Risikoelement	Monatsrang Fortschritt			
	Dieser Monat	Letzter Monat	Anzahl Monate	bei der Risikoüberwindung
Ersetzen des Entwicklers für die Sensorkontrollsoftware	1	4	2	Gewünschter Ersatzkandidat nicht verfügbar
Auslieferung der Zielhardware verzögert	2	5	2	Verzögerungen beim Beschaffungsverfahren
Datenformat für die Sensoren undefiniert	3	3	3	Aktionen des Software- u. Sensorteams nötig; fällig nächsten Monat
Personal für die Qualitätssicherung	4	2	3	Schlüsselperson verpflichtet, Fehlertoleranzprüfer benötigt
Fehlertoleranz gefährdet Leistung	5	—	3	Fehlertoleranzprototyp war erfolgreich
Datenbusänderungen berücksichtigen	6	—	1	Treffen der Datenbusentwerfer terminiert
Schnittstellendefinitionen für die Testumgebung	7	8	3	Einige Verzögerung bei den Aktionen; Treffen terminiert
Unsicherheiten in der Benutzeroberfläche	8	6	3	Prototyp erfolgreich
Betriebskonzept erstellen	—	7	3	erledigt
Unsicherheiten in der wiederverwendeten Überwachungssoftware	—	9	3	geforderte Entwurfänderungen erfolgreich durchgeführt

## ■ Risikomanagement

- Einleitung
- Risikomanagement-Prozess
- Unsicherheiten und Risikodiagramme
- Nutzwert vs. Risiko
- 5 Kernrisiken einer Softwareentwicklung
- Top Ten-Risikoliste
- **Checkliste Risikomanagement**
- Risikomanagement in Scrum

# Checkliste Risikomanagement

Prüfen Sie das Risikomanagement für Ihr(e) Projekt(e) anhand folgender Checkliste:

1. Gibt es eine Risikoliste, die alle Kernrisiken von Softwareentwicklungsprojekten und alle projektspezifischen Risiken enthält?
2. Ist ein kontinuierlicher Risikoidentifizierungsprozess etabliert, der eine offene Diskussion potenzieller Projektrisiken fördert und ermöglicht?
3. Sind Unsicherheitsdiagramme vorhanden, die genutzt werden, um Kausalrisiken zu quantifizieren sowie Gesamtrisiken und Erwartungen zu kommunizieren?
4. Gibt es Zielvorgaben, die sich am Nano-Prozent-Leistungsniveau orientieren, und Lieferabschätzungen, die das Unsicherheitsdiagramm berücksichtigen?

# Checkliste Risikomanagement

5. Ist jedem Risiko ein Eintrittsindikator zugeordnet, der kontinuierlich überwacht wird, um den Risikoeintritt festzustellen?
6. Gibt es für jedes Risiko einen Eventualfallplan und einen Risikoverminderungsplan?
7. Wird für jedes Risiko die Risikohöhe evaluiert?
8. Gibt es quantifizierte Nutzwerteinschätzungen und eine entsprechende Priorisierung der Produktkomponenten?
9. Falls die Softwareentwicklung inkrementell erfolgt: werden Informationen über Zeit, Aufwand und relative Größe für jede fertiggestellte Version festgehalten und ab der Mitte bis zum Ende des Projekts als Maß für den Fertigstellungsgrad genutzt?

## ■ Risikomanagement

- Einleitung
- Risikomanagement-Prozess
- Unsicherheiten und Risikodiagramme
- Nutzwert vs. Risiko
- 5 Kernrisiken einer Softwareentwicklung
- Top Ten-Risikoliste
- Checkliste Risikomanagement
- Risikomanagement in Scrum

# Scrum Risikomanagement

- Das Product Backlog nach Risiken zu priorisieren, erscheint einleuchtend

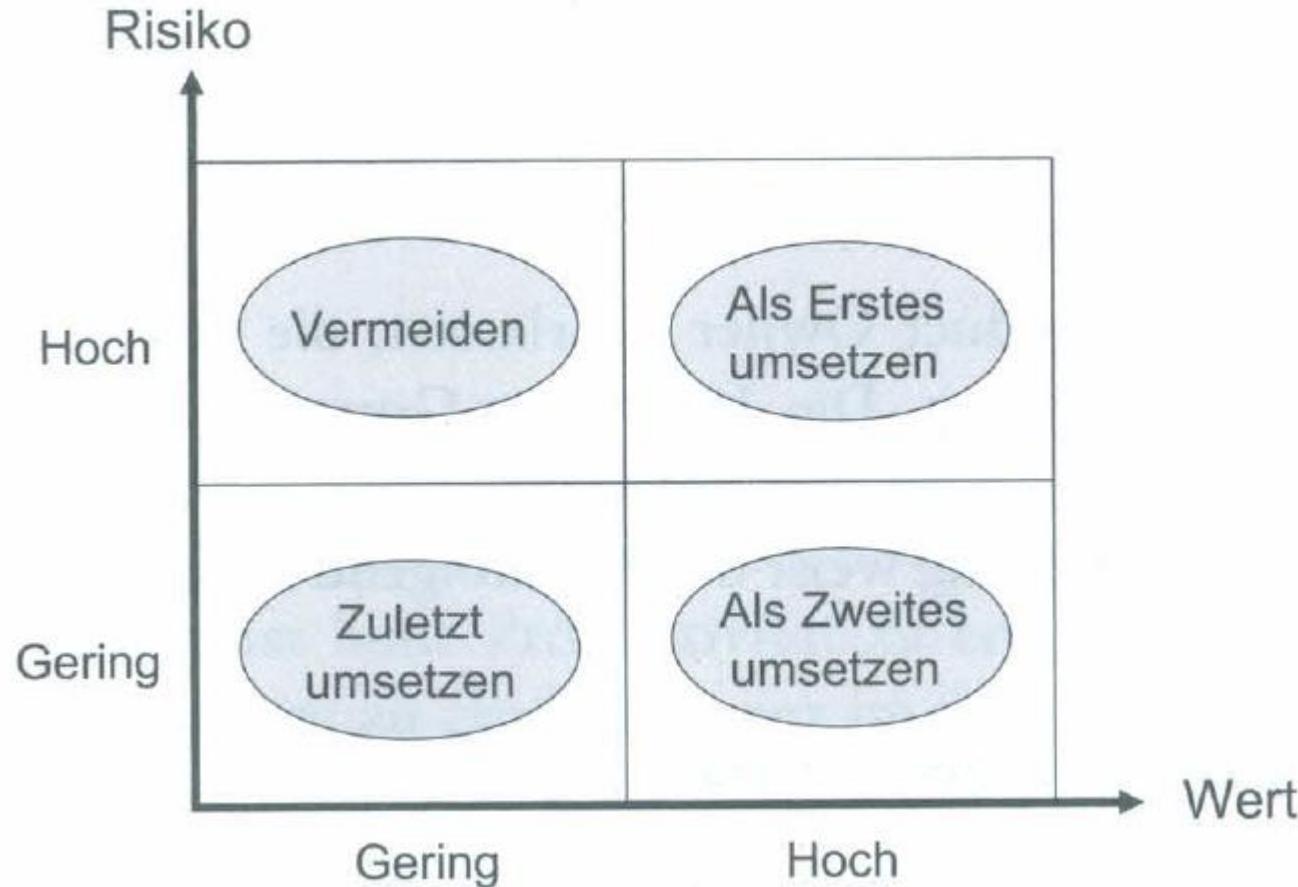
Priorität	Thema	Beschreibung	Akzeptanzkriterien	Aufwand
1	Kalender	Als Basisanwender möchte ich eine Besprechung anlegen.	Teste das Eingeben ungültiger Werte, z.B. Endzeit liegt vor Startzeit.	1
2	Kalender	Als Basisanwender möchte ich eine Besprechung stornieren.	Teste, dass dieselbe Besprechung nicht zweimal gelöscht werden kann.	3
3	Kalender	Als Basisanwender möchte ich eine Besprechung ändern.	Teste, ob die Änderungenpersistiert wurden.	2
...				

- Allgemein gilt: je innovativer Ihr Scrum-Projekt ist, desto mehr Unsicherheiten und Risiken weist das Projekt auf.

# Risiken analysieren

- Identifizierte Risiken sind zu analysieren:
  - Risikoursachen und Eintrittswahrscheinlichkeit werden ermittelt
  - Potenzieller Schaden bei Risikoeintritt wird geschätzt
  - Geeignete Gegenmaßnahmen werden identifiziert und als neue Einträge im Product Backlog eingefügt
  - Achtung: Zur Ermittlung der richtigen Gegenmaßnahmen sind die aufgefundenen Risikoursachen besonders wichtig
  - Exemplarische Beispiele für mögliche Gegenmaßnahmen:
    - Erstellen von Durchstich-Prototypen, um technische Risiken zu minimieren
    - Durchführung von Trainings- und/oder Schulungsmaßnahmen, um Know-how aufzubauen und Engpässe zu vermeiden

# Wert-Risiko-Matrix (nach Cohn)



# Wert-Risiko-Matrix (nach Cohn)

- Anforderungen, die einen hohen Wert UND ein hohes Risiko beinhalten, werden mit hoher Priorität klassifiziert und als Erstes realisiert
  - Hoher Nutzen wird rasch realisiert und schädliche Risiken werden frühzeitig adressiert
- Anforderungen, die einen hohen Wert und ein geringes Risiko beinhalten, werden mit mittlerer Priorität klassifiziert
- Anforderungen, die einen geringen Wert und ein geringes Risiko beinhalten, werden mit niedriger Priorität klassifiziert
- Und: Umsetzung von Anforderungen mit geringem Wert aber hohem Risiko sollte vermieden werden!

**Herzlichen Dank für  
Ihre Aufmerksamkeit !**