

# Risikomanagement – Was ist Risiko?

- Risiko ist die **Wahrscheinlichkeit**, dass ein Schaden bestimmter Höhe eintritt
  - 0**: Schaden tritt nie ein
  - 1**: Schaden tritt immer ein
- Das Risiko steigt mit der **Eintrittswahrscheinlichkeit**  $W$  der Gefahr
- Das Risiko steigt mit der **Schadenshöhe**  $S$
- **Risikofaktor**  $\Gamma$  ist das Produkt von Eintrittswahrscheinlichkeit und Schadenshöhe

$$\Gamma =_{\text{def}} W \cdot S$$

- Ein Risikofaktor wird somit in der Einheit geführt, in der die Schadenshöhe gemessen wird

## Risikomanagement (Forts.)

### Die Top Ten der Risiken bei der SW-Entwicklung

1. **Personelle** Defizite  
Choleriker, Unerfahrenheit, Desinteressiertheit, ...
2. Unrealistische Termin- und Kostenvorgaben  
Wunschtermin wird vom Vertrieb ohne Rückfrage festgelegt, ...
3. Entwicklung von falschen Funktionen, Eigenschaften  
Anwender ist falsch verstanden oder desinteressiert, ...
4. Entwicklung von **unbrauchbaren GUIs**  
Zu viele Freiheitsgrade, zu Text lastig, ...
5. **Vergolden** (über das Ziel hinaus)  
Unerwartete Features werden kreiert, ...

# Risikomanagement (Forts.)

## Risikomanagement besteht aus sechs Schritten

1. Risiko **Identifikation**
  - Ergebnis: Risikoelemente, die Projekterfolg gefährden
  - Kann über Checklisten ermittelt werden
2. Risiko **Analyse**
  - Risikofaktor = Wahrscheinlichkeit x Schadensausmaß
3. Risiko **Prioritätenbildung**
4. Risiko **Management Planung**
  - Risikomanagement Plan für jedes Risikoelement in Projektplan integrieren
5. Risiko **Überwindung**
6. Risiko **Überwachung**
  - Festlegung von Überprüfungsterminen
  - Berichte über Fortschritt der Risikoelemente
  - Konzentration auf die Beseitigung Risikoelemente

## Risikomanagement (Forts.)

### Die Top Ten der Risiken bei der SW-Entwicklung

6. Kontinuierliche **Anforderungsänderungen**  
Zu niedrige Änderungsschwelle, ...
7. Defizite bei extern gelieferten Komponenten  
Fehlende Eingangstests und Kompatibilitätsüberwachungen, ...
8. Defizite bei externen Aufträgen  
Können vermieden werden durch: Bindungen an Erfolg, Leistungstest, Inspektionen, Kompatibilität
9. Defizite in der Echtzeitleistung  
Vermeiden durch Simulation, Prototypen, Leistungstest, Instrumentierung, Modellierung, Tuning
10. **Überfordern** der Softwaretechnik  
Fehlende Prototypen, Aufwand und Ergebnis sind unverhältnismäßig, ...