TECHNISCHE UNIVERSITÄT DRESDEN

Department of Computer Science Institute of System Architecture, Operating Systems Group

DIE MATHEMATIK DES VERBERGENS

HERMANN HÄRTIG,
CLAUDE-JOACHIM HAMANN,
MICHAEL ROITZSCH

TECHNISCHE UNIVERSITÄT DRESDEN

DISCLAIMER

Ich erzähle über…

- das Finden von **Sicherheitslücken**
- unser **mathematisches Modell** dafür
- den **Vergleich** von offen und geschlossen

Ich erzähle nicht…

- ob open oder closed source **besser** ist

TECHNISCHE UNIVERSITÄT DRESDEN

BATTLE ROYALE

Open Source

Closed Source

TECHNISCHE UNIVERSITÄT DRESDEN

BATTLE ROYALE

Open Source	Closed Source
jeder hat Zugriff zum Quellcode	nur die Firma hat den Quellcode
jeder kann nach Fehlern suchen	Angreifer haben es schwerer
mehr Verteidiger finden mehr Fehler	Ausnutzen der Fehler wird erschwert
Fehler finden leichter	**Fehler finden schwerer**

HYPOTHESE

Angreifer brauchen nur einen Fehler.
Verteidiger müssen alle Fehler finden.

MODELL

3 Fehler:

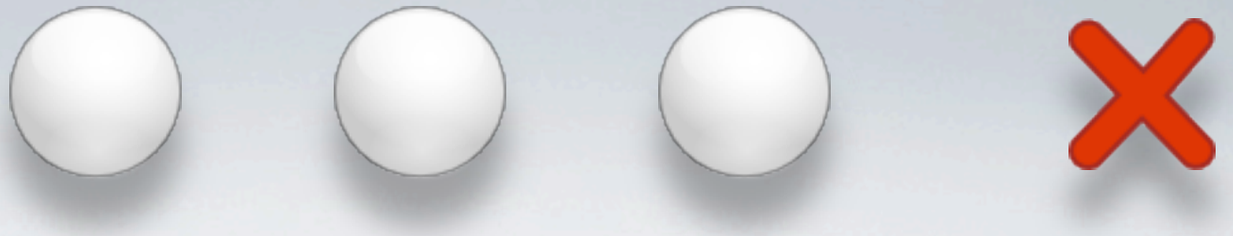
$e = 3$

Kein Fehler:

p, q

ANGREIFER

$a = 3$

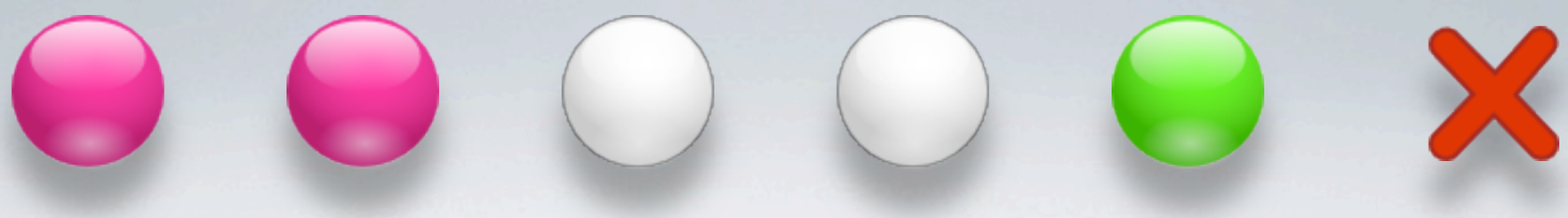


ANGREIFER

$$p_A = 1 - (1 - ep)^a$$

VERTEIDIGER

$d = 5$



TECHNISCHE UNIVERSITÄT DRESDEN

VERTEIDIGER

$$p_D = e! \cdot \sum_{i=0}^{d-e} \binom{d}{i} q^{d-i} \left(1 - eq\right)^i S_{d-i,e}$$

BEISPIEL

- 20 Fehler

$$e = 20$$

- 1% Fehler-finde-Wahrscheinlichkeit

$$p = q = 0.01$$

- 75% geforderte Gewinnchance

$$p_A = p_D = 0.75$$

- Wie viele Angreifer?

$$a = 7$$

- Wie viele Verteidiger?

$$d = 424$$

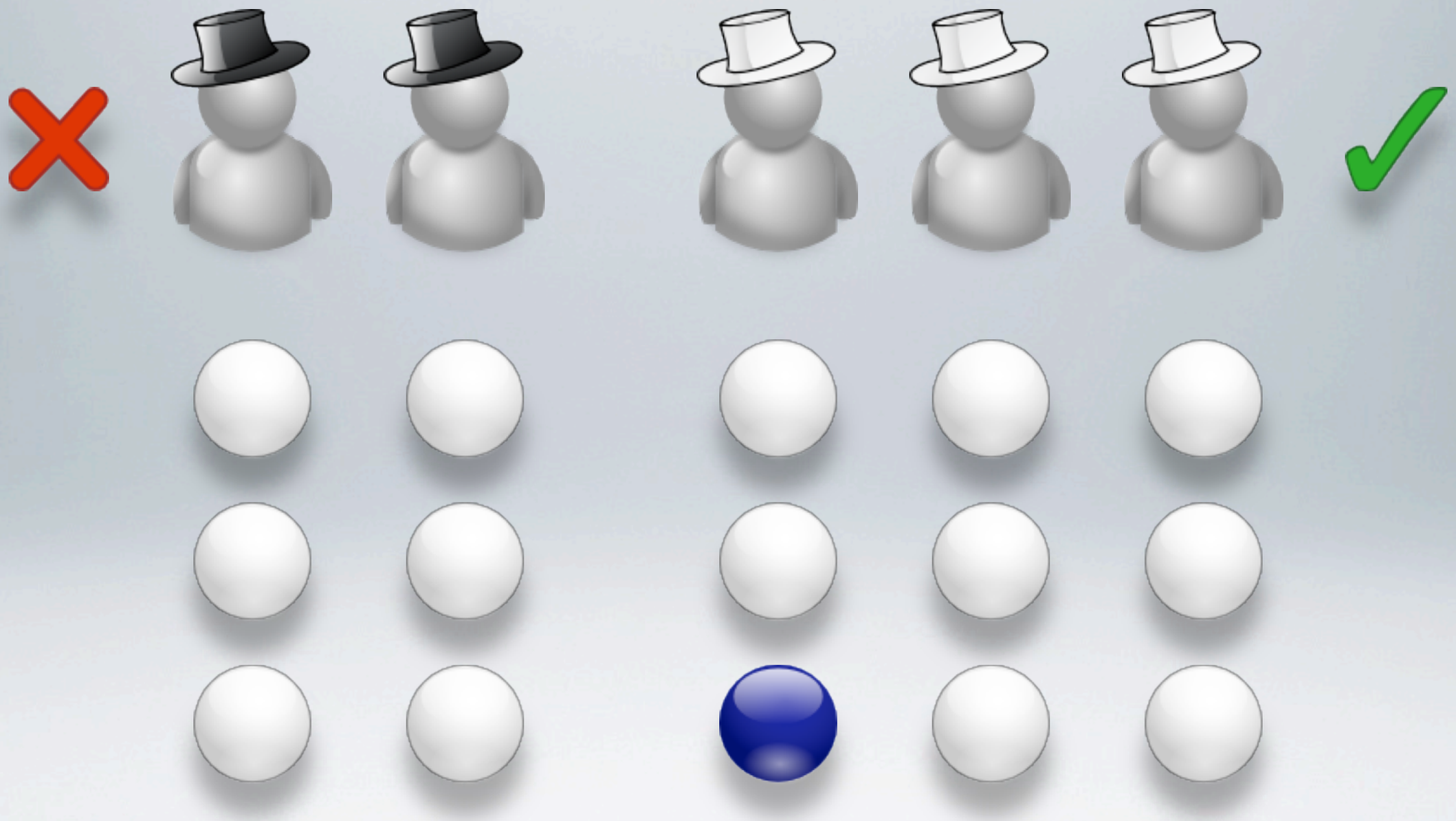
GEMECKER

- Was ist wenn beide Seiten verlieren?
- … oder gewinnen?
- Verlieren die Verteidiger wirklich, wenn sie nicht alle Fehler finden?
- Sie müssen die Fehler nur **zuerst** finden.
- anstelle eines Schnappschusses als **Wettlauf** modellieren

HYPOTHESE

Verteidiger müssen jeden Fehler vor den Angreifern finden.

WETTLAUF



BERECHNUNG

p m Schritte

q n Schritte

$$p_{m,n} = (1-p)^{m-1}\, p \cdot (1-q)^{n-1}\, q$$

BERECHNUNG

$$p_{m,n} = (1-p)^{m-1}\, p \cdot (1-q)^{n-1}\, q$$

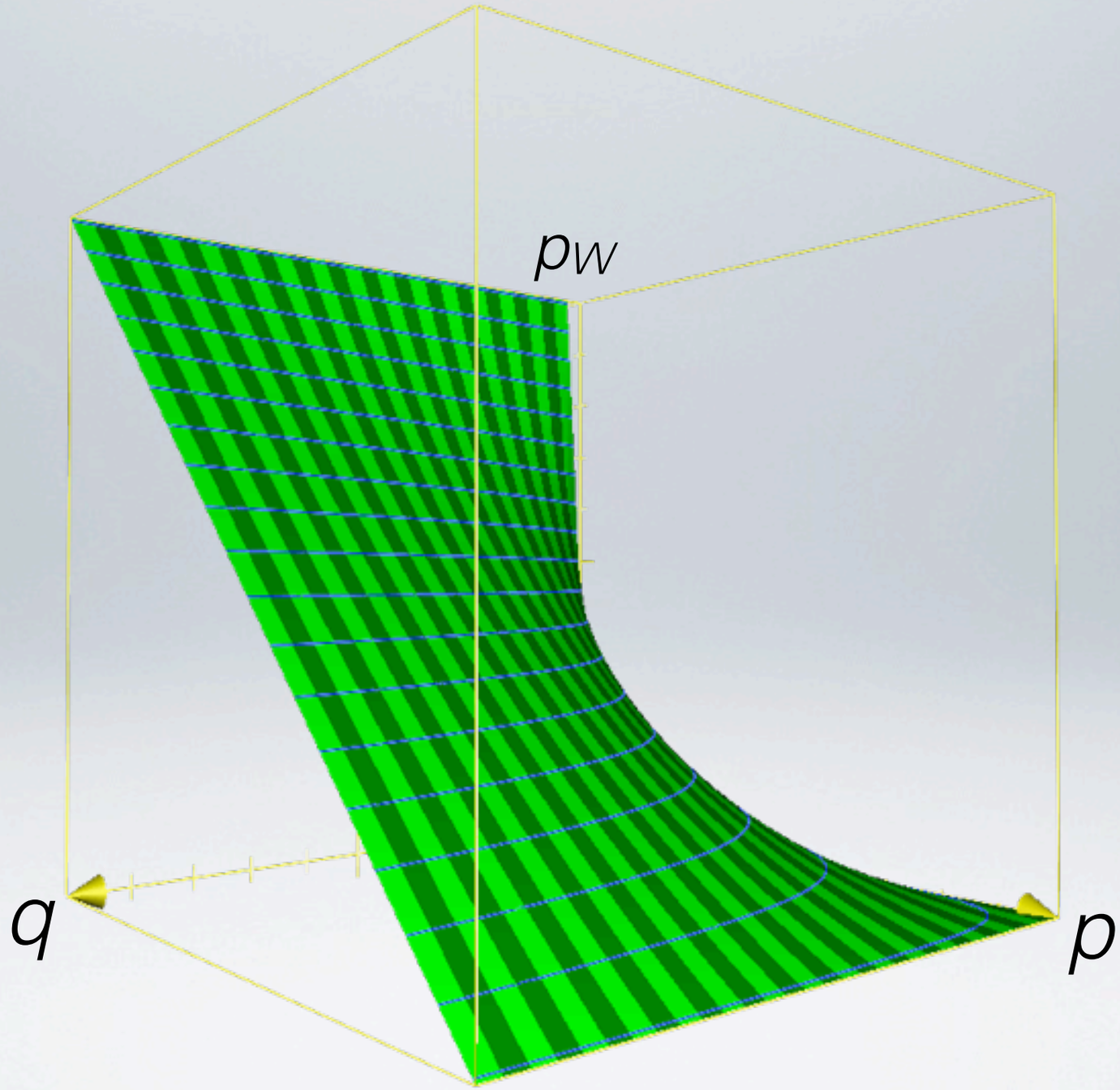
defenders win for $n < m$

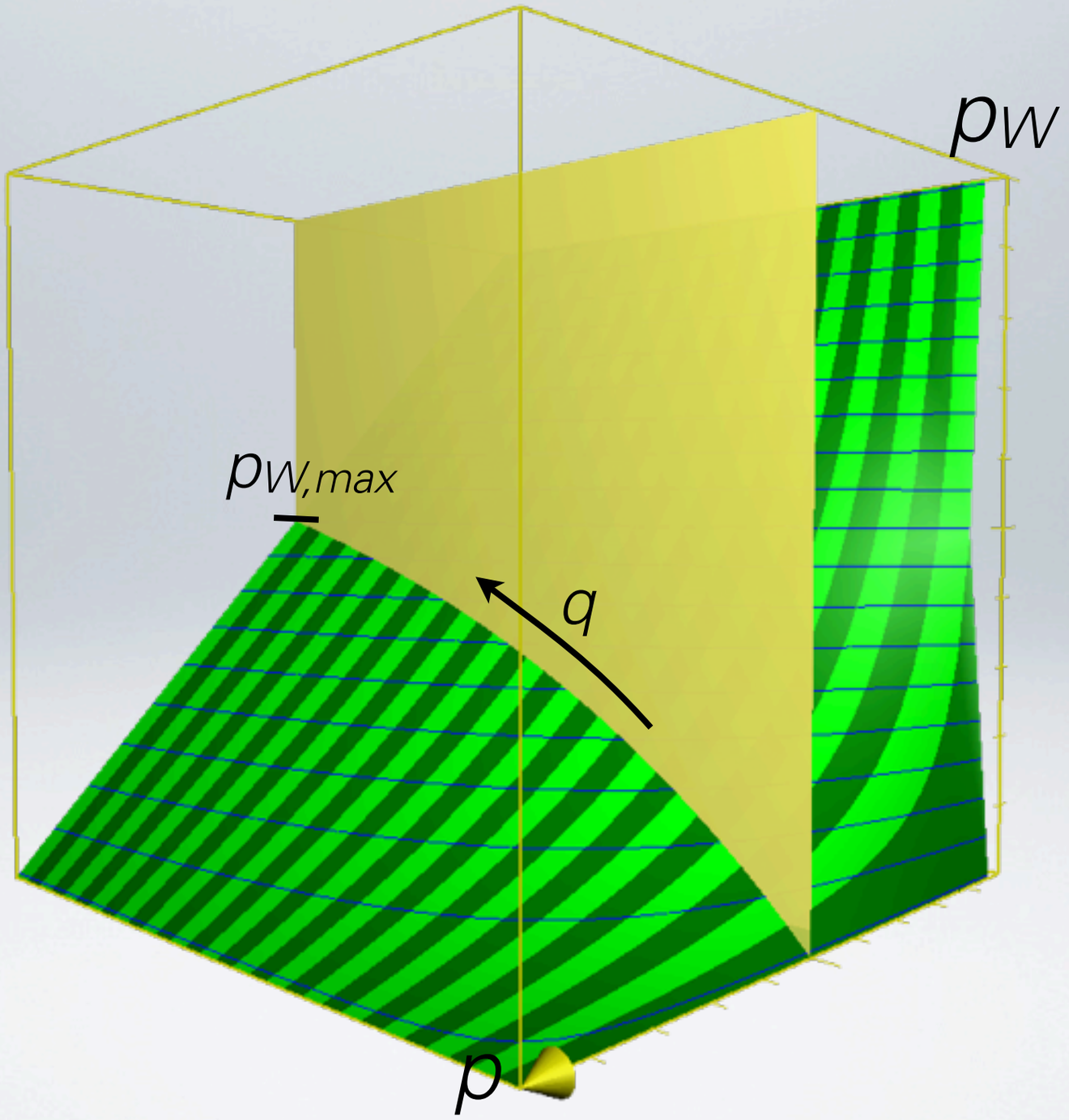
$$p_W = \sum_{n=1}^{\infty} \sum_{m=n+1}^{\infty} p_{m,n} = \frac{q\,(1-p)}{q\,(1-p)+p}$$

BERECHNUNG

$$p_W = \sum_{n=1}^{\infty} \sum_{m=n+1}^{\infty} p_{m,n} = \frac{q\,(1-p)}{q\,(1-p)+p}$$

Open Source	**mehr Verteidiger**	**höheres q**
Closed Source	**schwerer für Angreifer**	**kleineres p**


pw
q
p


p_W
$p_{W,max}$
q
p

BEISPIEL

- 1 Million Codezeilen, 15 Sicherheitsfehler

$$e = 15$$

- Wahrscheinlichkeit für einen Verteidiger

$$q_{single} = 0.002\%$$

- für einen Angreifer im Open-Source-Fall

$$p_{single,open} = 0.002\%$$

- Closed Source Faktor 2 schwerer

$$p_{single,closed} = 0.001\%$$

- 500 Angreifer
- Wie viele Verteidiger brauchen wir?

ERGEBNIS

	pw = 0.6	pw = 0.9
Closed Source	7815	62088
Open Source	17133	unmöglich

FOLGERUNG

Egal wie viele Verteidiger, das Fenster ist für die Angreifer immer einen Spalt offen.

KURZFASSUNG

- **Urnenmodell** für das Finden von Sicherheitslücken
- **Wettlauf** zwischen Angreifern und Verteidigern
- es gibt eine **obere Schranke** für die Verteidiger
- diese Schranke wird in der Realität vielleicht erreicht