

Titel

INNOQ TECHNOLOGY DAY 2024 / 05.12.2024

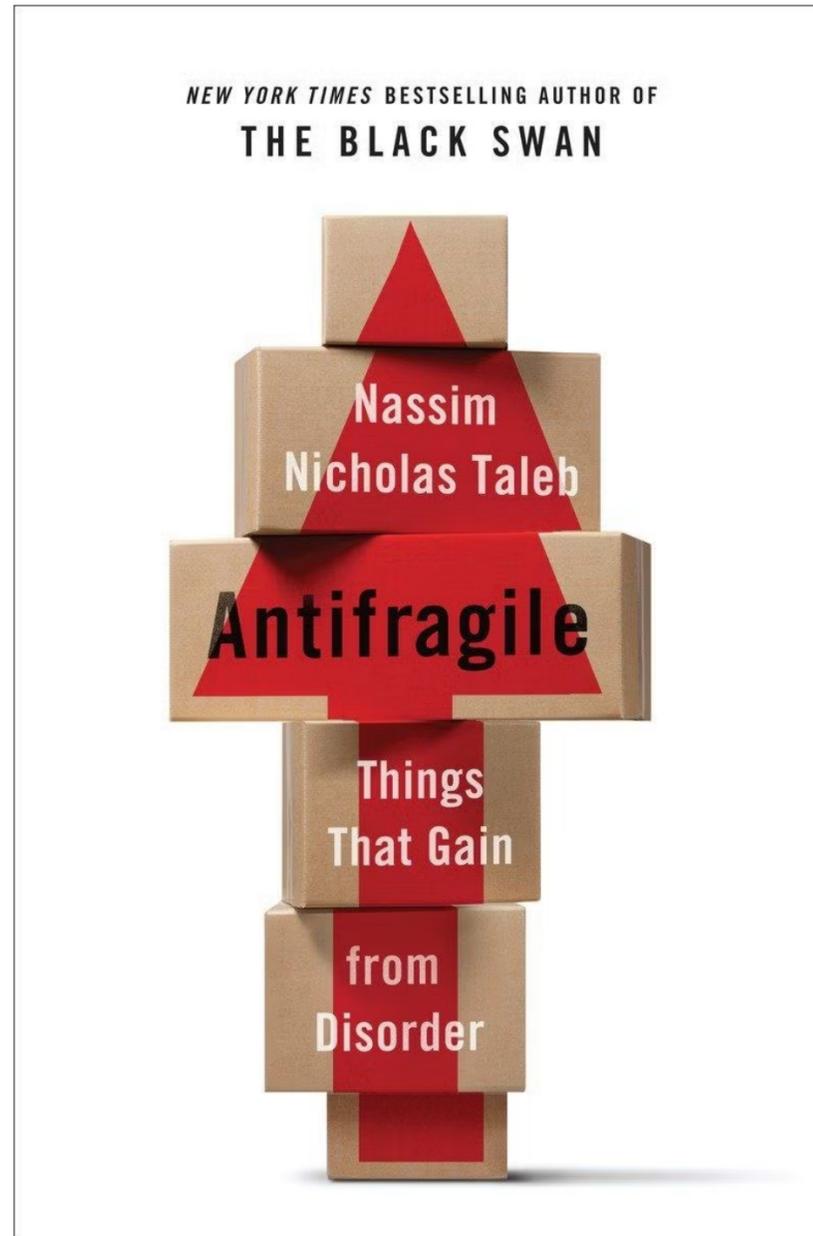
Antifragile Software- entwicklung

INNOQ



DANIEL WESTHEIDE
@KAFFEECODER

Antifragile (Taleb, 2012)



Antifragile: Things That Gain from Disorder
(Nassim Nicholas Taleb, 2012)

Was bedeutet überhaupt fragil?

fragil: die "Unfähigkeit, Unbeständigkeit zu vertragen" (Taleb, 2012)



Die Triade

Fragil

Robust

Antifragil



unfähig
Unbeständigkeit zu
vertragen

hält Unbeständigkeit
aus

profitiert von
Unbeständigkeit

Fragile Softwareentwicklung

- **Softwaresysteme:** anfällig gegenüber Hardwareausfällen oder Lastspitzen
- **Entwicklungsprozesse:** Alles muss nach Plan laufen
- **Softwarearchitekturen:** nach unerwarteten Anforderungen unpassend
- **Organisationen:** nicht in der Lage sich an den Markt anzupassen

**Was braucht es für ein
antifragiles
Softwareentwicklungssystem?**

Das Antifragile

Eigenschaften und Strategien

Asymmetrie

Unsicherheit, Variabilität und Stressoren...

...nutzen mehr als dass sie schaden



antifragil

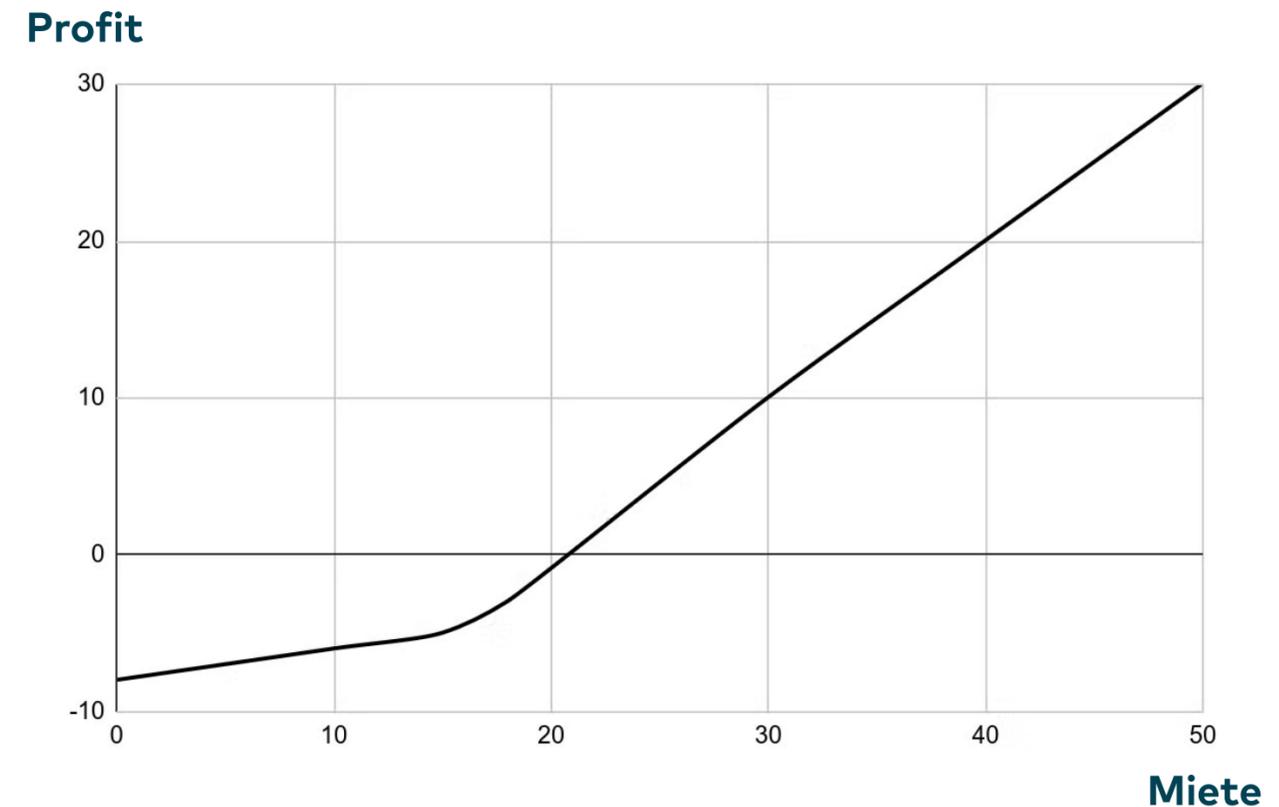
...schaden mehr als dass sie nutzen



fragil

Optionalität

- Das Recht, aber nicht die Pflicht etwas zu tun
- **Option = Asymmetrie + Rationalität**
- erste bekannte Option der Menschheitsgeschichte: Thales und die Nutzungsrechte für Olivenpressen
- Ersatz für Wissen



Antifragilität des Thales. Nachgebildet aus Antifragile (Taleb, 2012)

Schwarze Schwäne

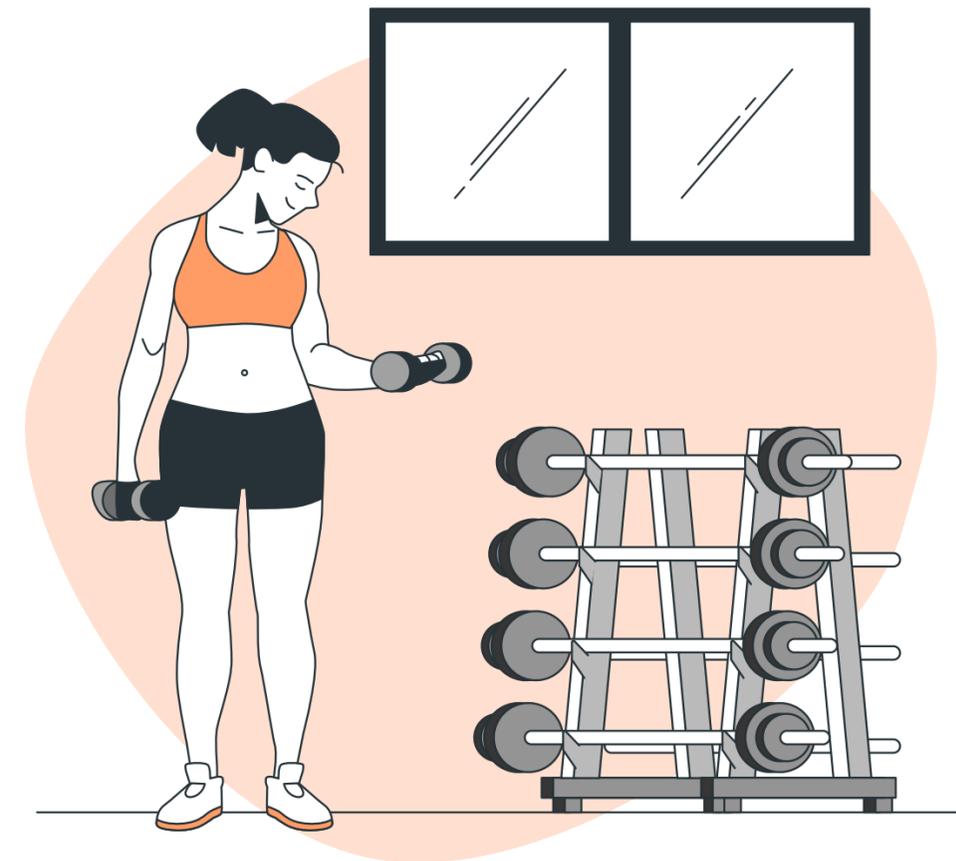


Foto von [Pawel Czerwinski](#) auf [Unsplash](#)

- irreguläre Ereignisse mit massiven Folgen
- keine Vorhersage oder Berechnung der Wahrscheinlichkeit möglich
- positive und negative Schwarze Schwäne

Hantelstrategie

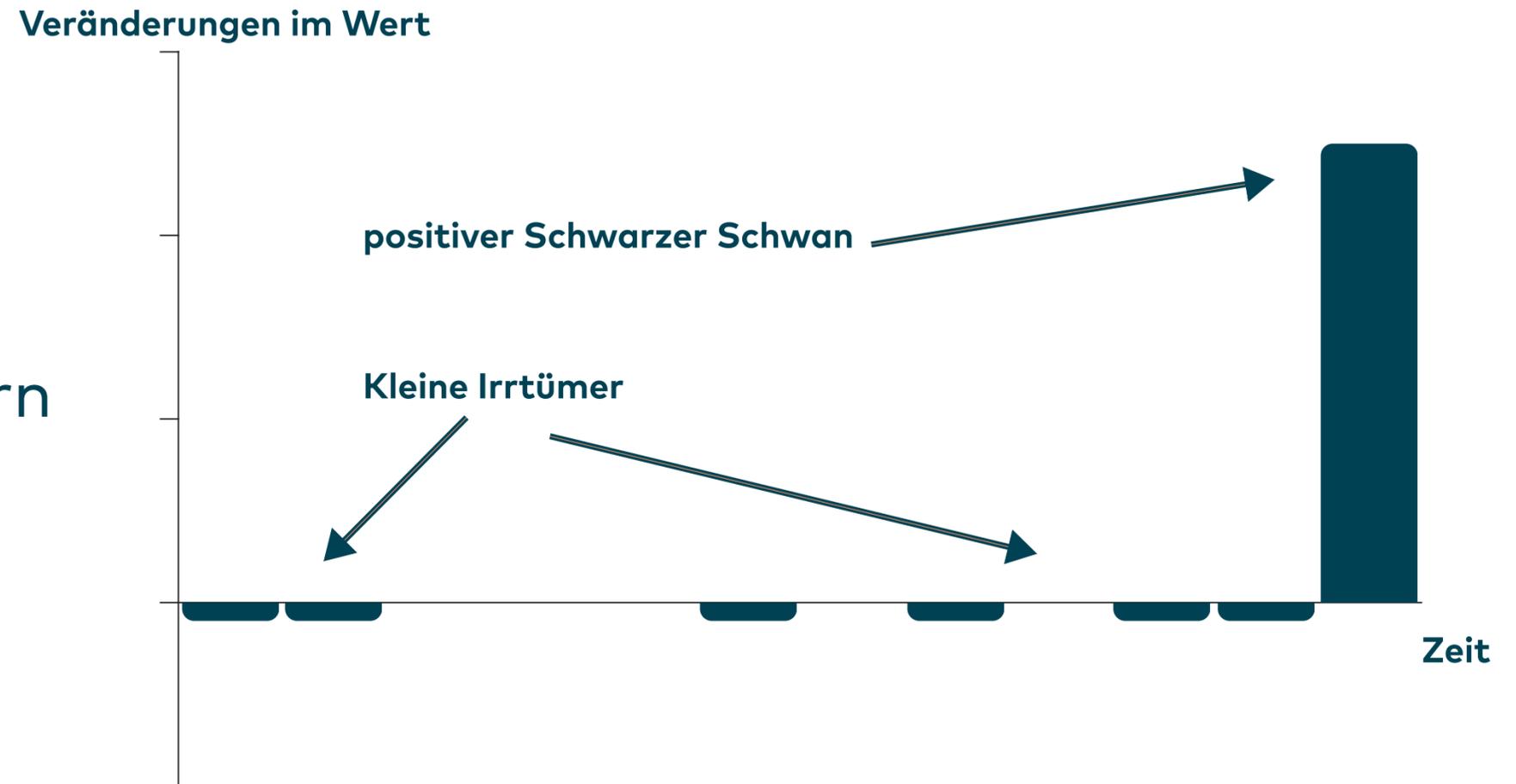
- robust gegenüber negativen Schwarzen Schwänen
- offen für positive Schwarze Schwäne
- Hantelform ergibt sich durch Besetzen der Extreme und Aussparen der Mitte



Quelle: [Weights Cartoon Illustrations auf Storyset](#)

Antifragiles Tüfteln

- viele kleine Irrtümer
- Aussicht auf positiven Schwarzen Schwan
- nicht komplett zufällig, sondern rational
- jeder Irrtum eine wertvolle Information
- frühes Scheitern

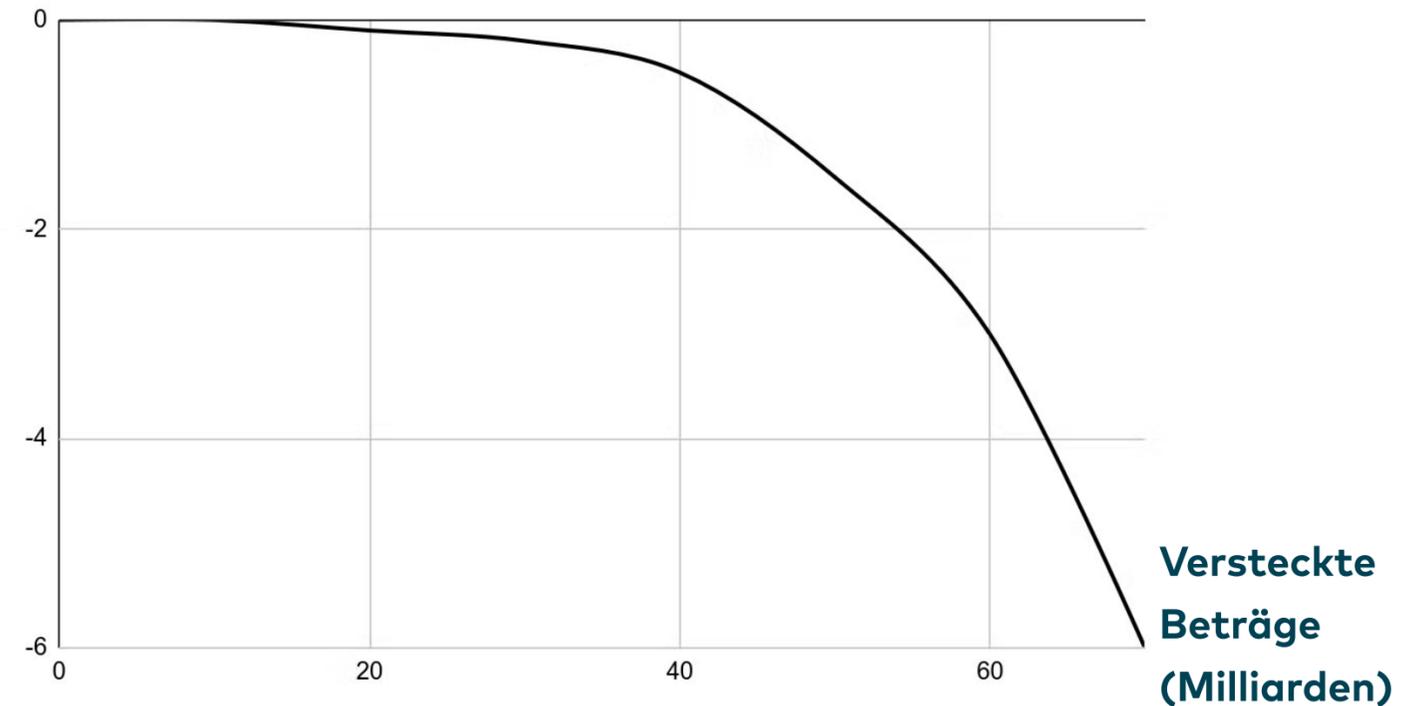


Mechanismus des optionalen Versuchs und Irrtums.
Nachgebildet aus Antifragile (Taleb, 2012)

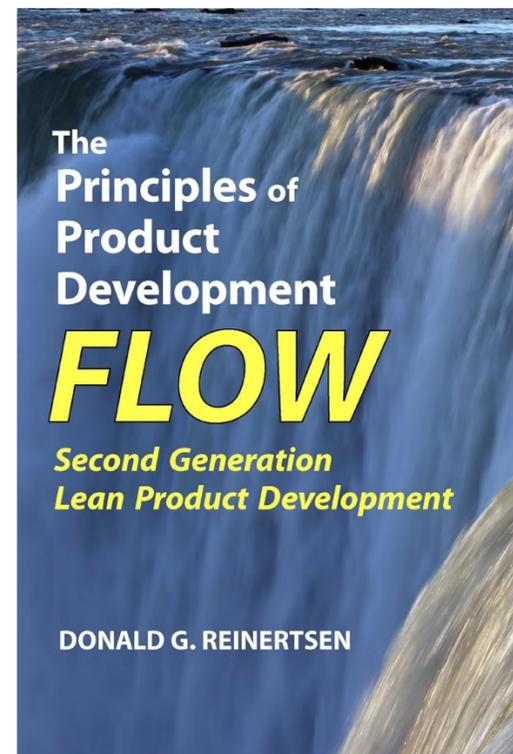
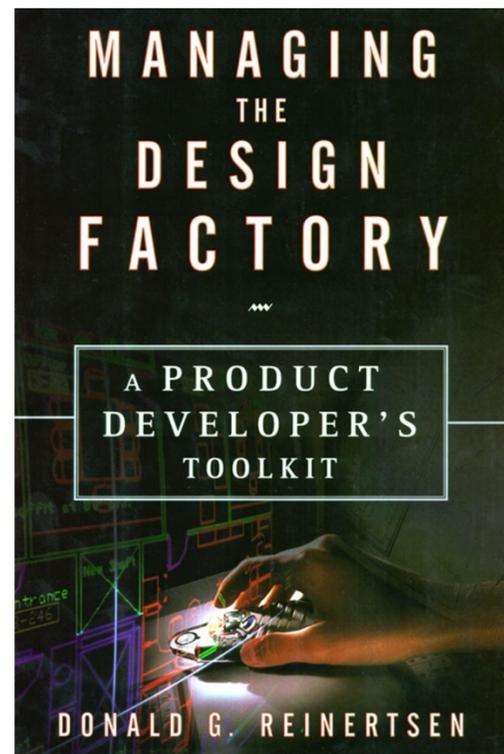
Bottom-up statt Top-down

- Das Große ist fragiler als das Kleine
- auch in der Summe als Ansammlung kleiner Einheiten
- Ein großes Unternehmen ist anfälliger für negative Schwarze Schwäne als mehrere kleine Unternehmen
- Nicht-Linearität

Liquidationskosten
(in Milliarden)



Antifragile Softwareentwicklung?



- Donald G. Reinertsen
- Managing the Design Factory (1997)
- The Principles of Product Development Flow (2009)
- Schwerpunkt: Design / Entwicklung physischer Produkte

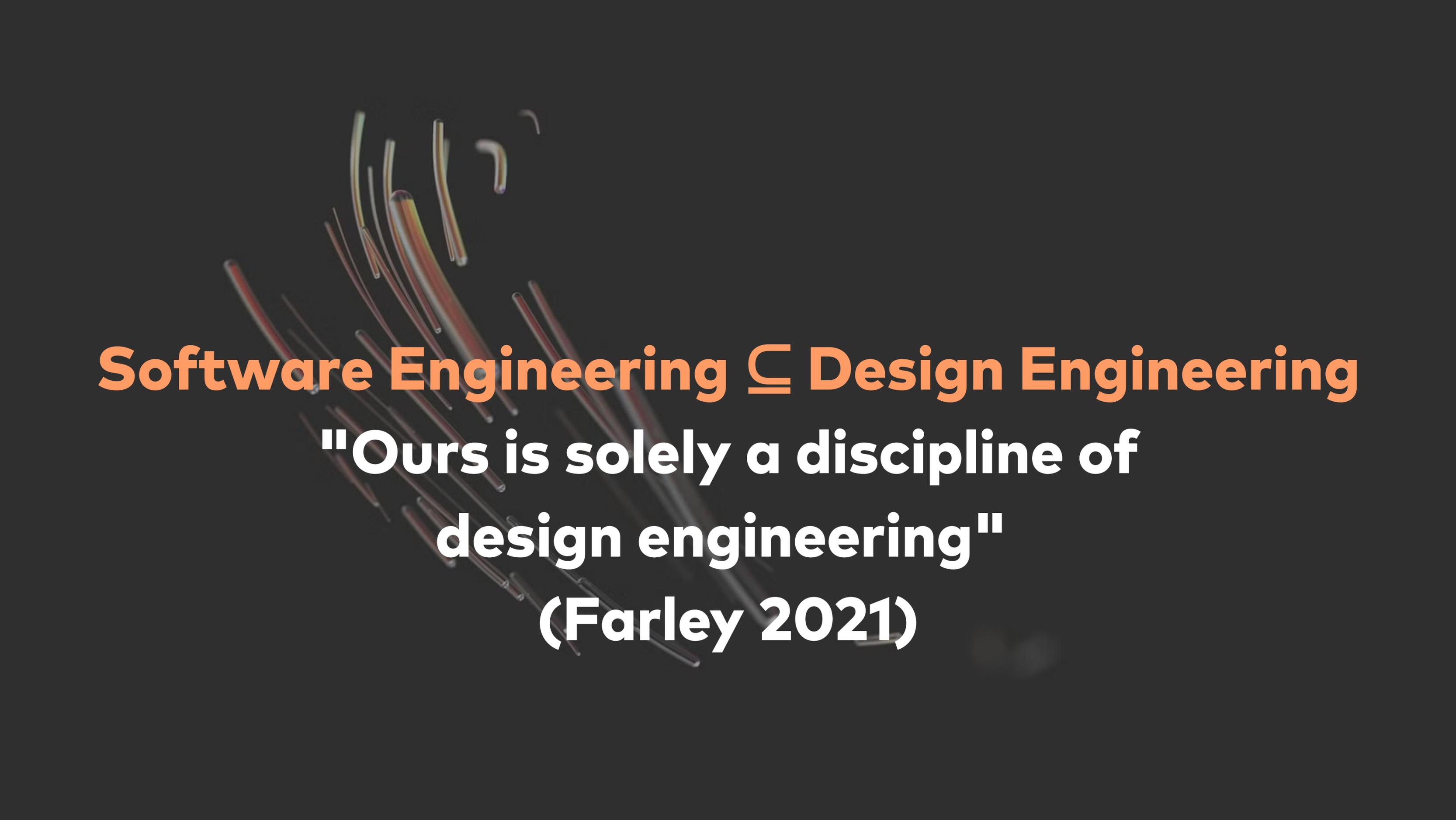
Design Engineering vs Production Engineering

Produktion

- Wiederholbarkeit
- Mehrmals das gleiche produzieren schafft Wert
- Variabilität minimieren

Design

- einmalige Prozesse
- Nur Veränderung schafft Wert
- Variabilität inhärent



Software Engineering \subseteq Design Engineering

**"Ours is solely a discipline of
design engineering"**

(Farley 2021)

Rationale Entscheidungen

statt Intuition

**We want information...
information... information**

The Prisoner (1967)

Informationsgewinn maximieren

- Bei Tests am meisten Information dort, wo die Erfolgswahrscheinlichkeit bei 50 Prozent liegt
- Informationstheorie: Je wahrscheinlicher ein Ereignis, desto weniger Information enthält es
- Explorative Tests sollten 50 Prozent Erfolgswahrscheinlichkeit anstreben
- Ziel: Lernen, Unsicherheiten und Risiken eliminieren

Fast Feedback over Zero Defects

- **Iterationsgeschwindigkeit** erhöhen günstiger als Fehlerrate zu minimieren
- Schlechte Pfade frühzeitig beenden
 - User Research
 - Spikes / technische Prototypen
 - Usability-Tests
 - Strategisches DDD: Domain Experts einbeziehen
- Experimenteller Ansatz der Softwareentwicklung
 - Continuous Delivery
 - Kleine Batchgrößen
 - schnelle Auslieferung,
 - frühes Feedback durch Nutzende

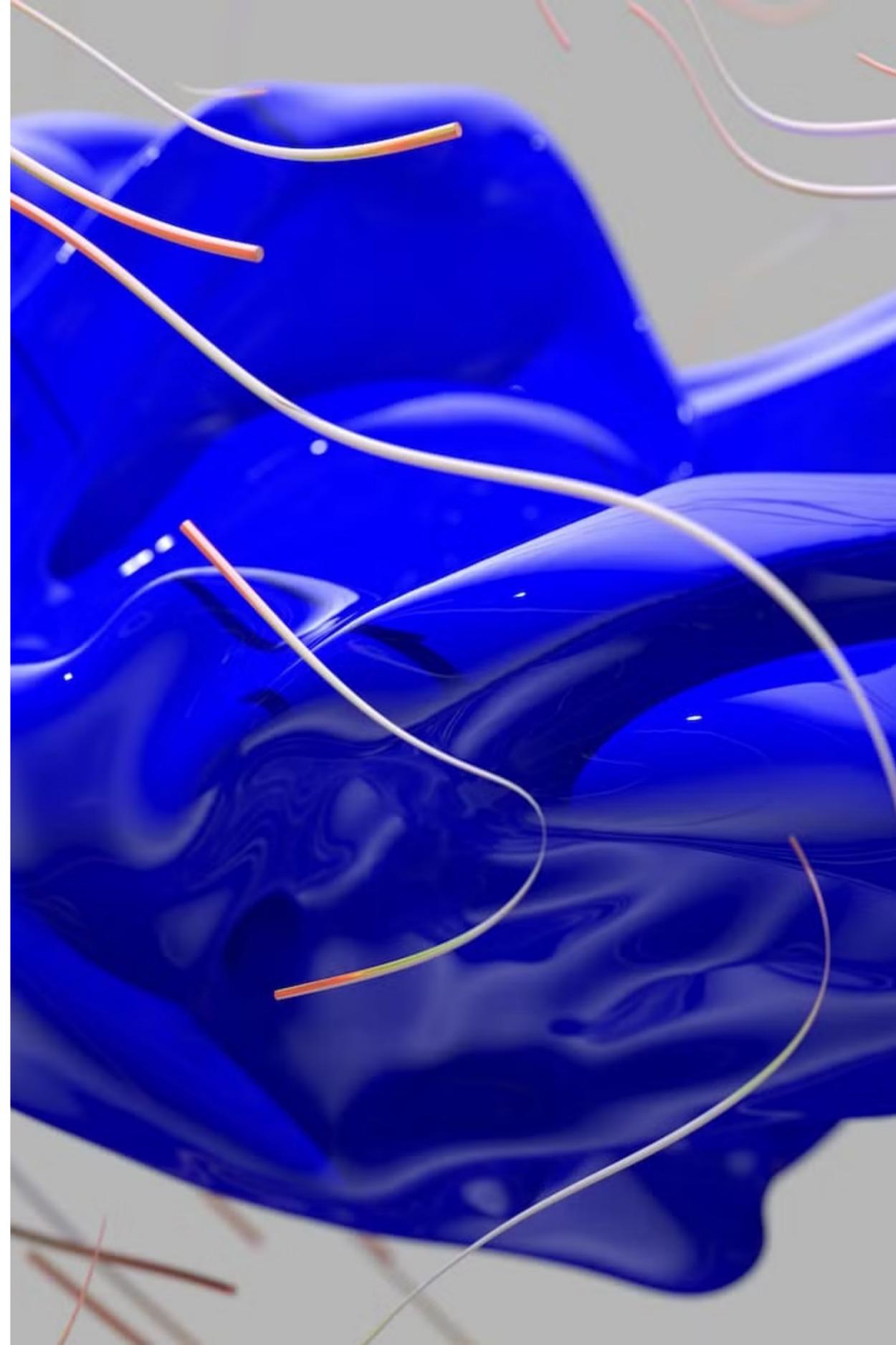
DORA State of Devops Report 2024

Performance level	Deployment frequency	Change fail rate
Elite	On demand (multiple deploys per day)	5%
Low	Between once per month and once every six months	40%

Auszug aus Performance Levels im DORA State of Devops Report 2024

Synthese

- Experimentelle, iterative Produktentwicklung ist antifragiles Tüfteln
- Spikes, User Research etc. setzen das Prinzip "frühes Scheitern" um und folgen der Hantelstrategie

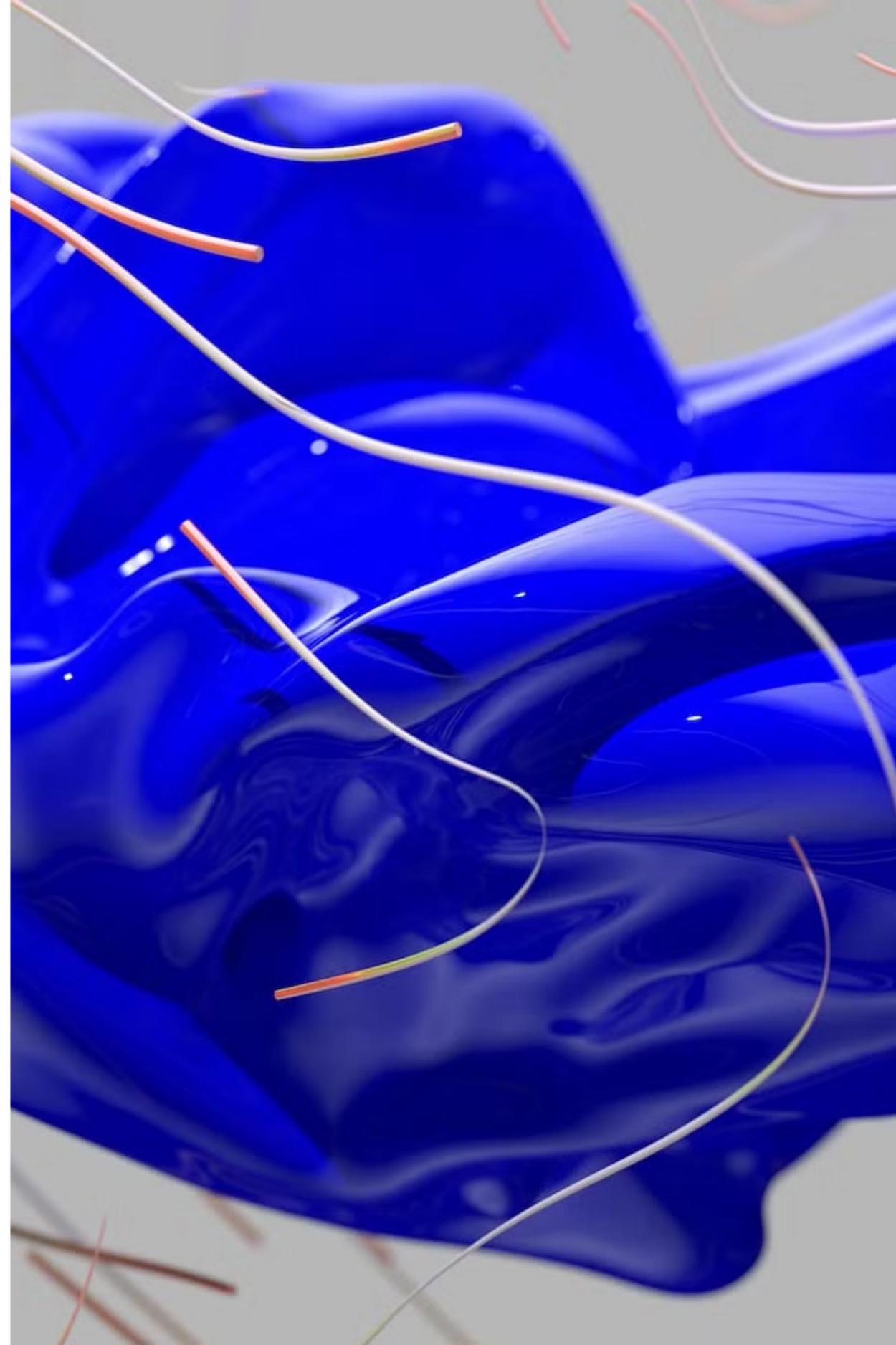


Aus Fehlern lernen

- Information aus Fehlschlägen so wertvoll wie Information aus Erfolgen
- Fehlschläge wünschenswert, wenn sie neue Information generieren
- Unwahrscheinliche Fehlschläge haben hohen Informationsgehalt
- Aufbau von Domänenexpertise im Zeitraffer
- Fehlschläge zelebrieren, wenn ihre Möglichkeit mit besonnenem Risiko in Kauf genommen wurde

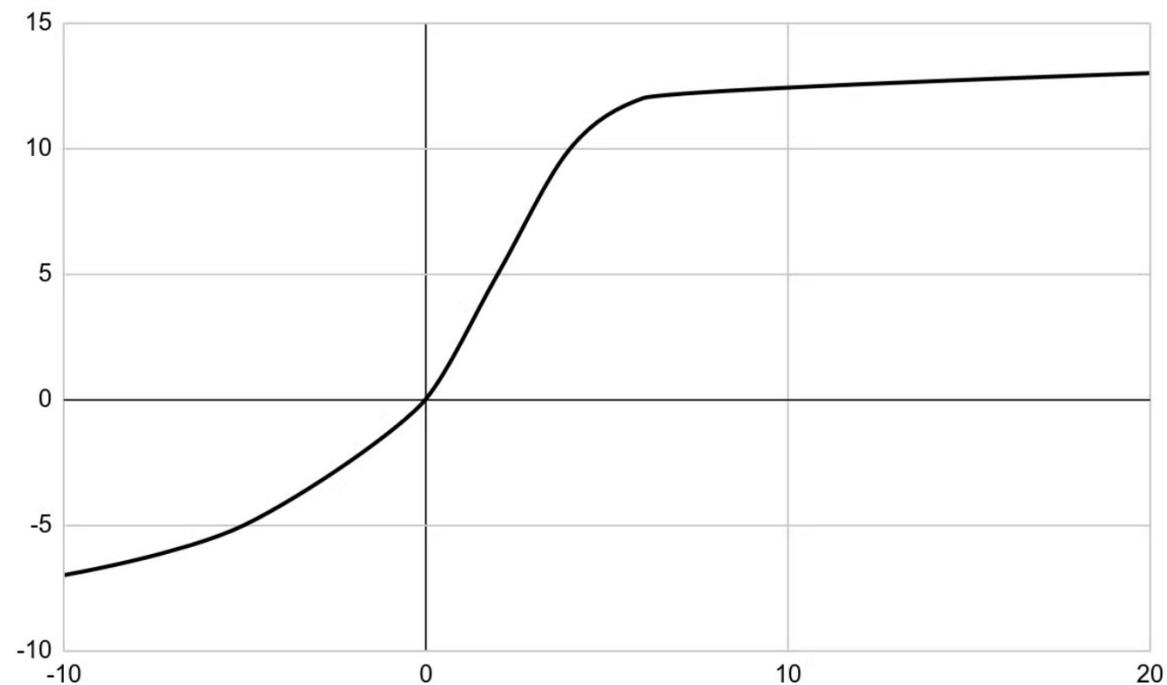
Synthese

- Reinertsen (1997) formuliert Kern-Aspekte der DevOps-Bewegung
- Wer antifragil sein will, schätzt Fehler als Informationsquelle
- Das Antifragile lernt auch aus den Fehlern der anderen
- Unwahrscheinliche Fehlschläge als Schwarze Schwäne mit hohem Erkenntnisgewinn



Variabilität ausnutzen

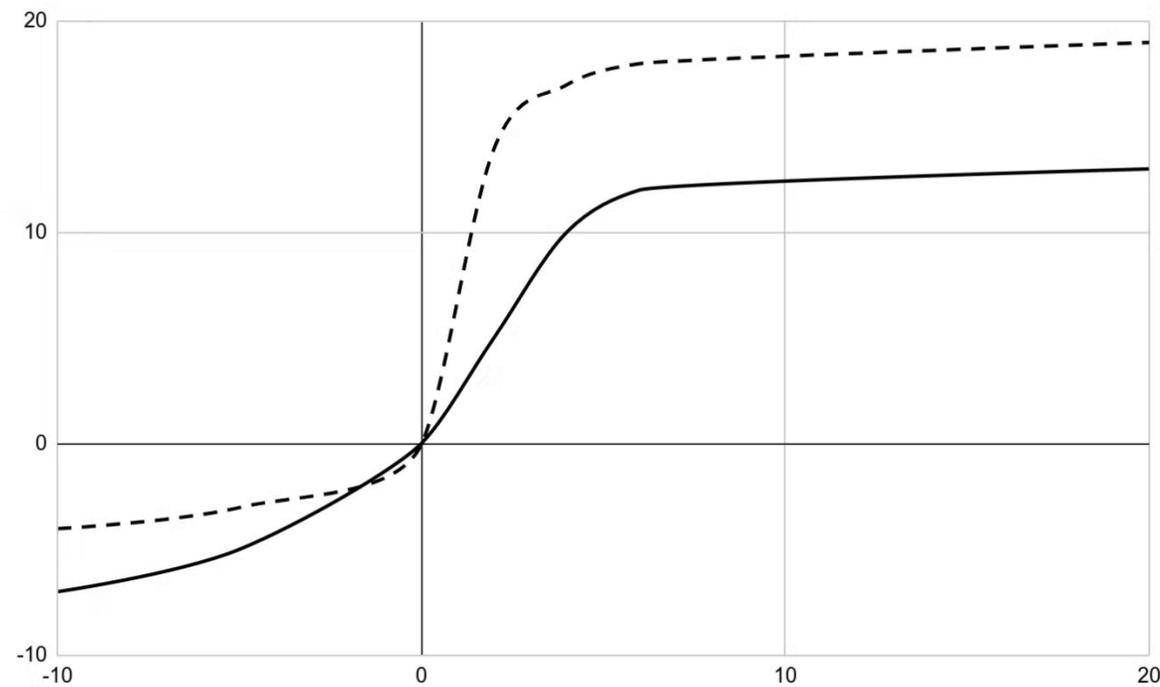
Nutzenfunktionen in der Produktentwicklung



- unbegrenzte Verluste, begrenzter Nutzen
- Optimum an Variabilität ist nicht maximale Variabilität

Product Development Payoff-Function. Nachgebildet aus Reinertsen (2009)

Die Nutzenfunktion ändern



Product Development Payoff-Function. Nachgebildet aus Reinertsen (2009)

- schlechte Pfade früh beenden
- Over-delivery bei unerwartet einfachen Features
- Evolutionäre Architektur
- In beiden Fällen hilft schnelles Feedback

Technology Shelf

- Technologie- und Produktentwicklung entkoppeln
- unterschiedliche Variabilität und Unsicherheit
- Technologieentwicklung vom kritischen Pfad nehmen
- In der Softwareentwicklung:
 - 20-Prozent-Zeit
 - Team Topologies: Enabling Teams und Platform Teams

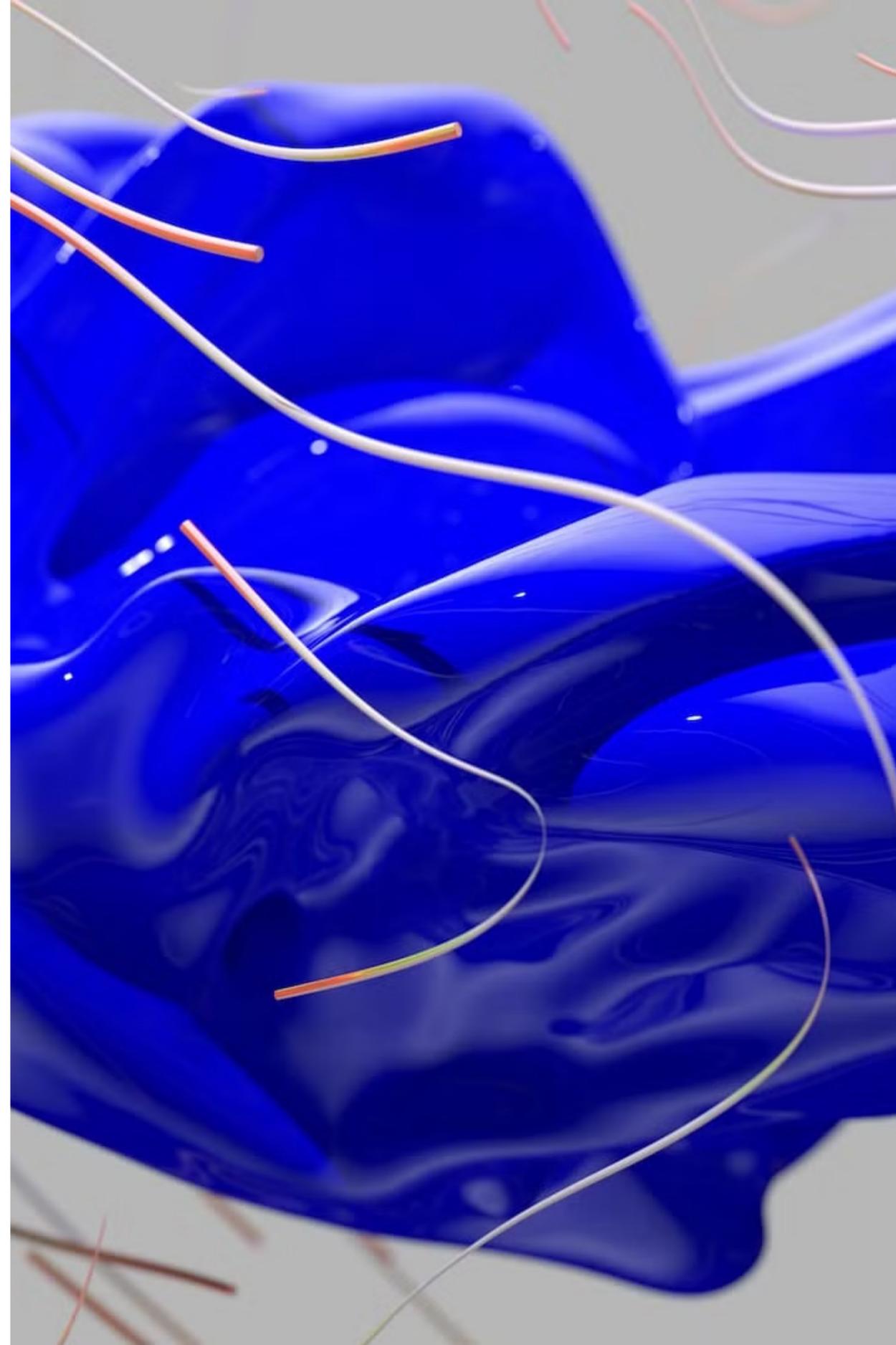
Synthese

- Taleb:
 - Optionen = Asymmetrie + Rationalität
 - Wer Optionen hat ist antfragil und profitiert von Unsicherheit
 - Wenn alles nach Plan laufen muss, werden Chancen, die sich auftun, verpasst
- Reinertsen:
 - häufig asymmetrische Nutzenfunktionen, die von Variabilität profitieren
 - schnelles Feedback verändert die Asymmetrie
 - Abweichungen vom Plan erlauben es, Gelegenheiten zu ergreifen
- Technologie-Entwicklung entkoppeln = Tüfteln nach der Hantelstrategie

Kontrolle dezentralisieren

Autonomes Team?

- kleine, unabhängige, interdisziplinäre Teams
- Optimierung auf Geschwindigkeit
- lokale Entscheidungen
- Befugnisse und Verantwortlichen vorab klären



Entscheidungsregeln

- abgeleitet aus betriebswirtschaftlichem Projektmodell
- Entscheidungen kontrollieren, ohne sich daran zu beteiligen
- Optimierung auf Systemebene statt lokaler Optimierungen

Optimierung auf Systemebene



Foto von [Riccardo Balbo](#) auf [Unsplash](#)

Interne Preisgestaltung

- Preisgestaltung für interne Premium-Dienstleistungen einer knappen Ressource
- Teams spüren sowohl Vorteile als auch Kosten ihrer Entscheidungen
- Marktmechanismen greifen
- Teams nehmen Dienstleistung nur dann in Anspruch wenn es betriebswirtschaftlich sinnvoll ist

Synthese

- Bottom-up eher antifrägil, top-down fragil
- Durch schnelle, lokale Entscheidungen können sich ergebende Chancen ausgenutzt werden
- Organisation kann von positiven Schwarzen Schwänen profitieren
- Viele kleine Einheiten weniger fragil als ein großes Etwas
- Scheitert eines von vielen kleinen, unabhängigen Teams, ist dies verkraftbar

Betrieb und Antifragilität

- Design Engineers sind nicht für den Betrieb zuständig, Software Engineers schon
- In Systemen, die zu sehr beruhigt wurden, sind die Risiken unsichtbar und wachsen an
- Kommt es doch zu einem Ereignis, ist dessen Auswirkung meist viel größer als in Systemen, die immer wieder Stressoren ausgesetzt sind
- Chaos Engineering
 - deckt vorhandene Risiken und Unsicherheiten auf
 - macht uns weniger anfällig für negative Schwarze Schwäne

Zusammenfassung

- Das Antifragile profitiert von Asymmetrie, Optionalität und rationalem Tüfteln
- Fehlschläge als wertvolle Information
- Hantelstrategie um sich abzusichern und offen für große Gewinne zu sein
- Tüfteln: experimenteller Ansatz der Produktentwicklung
- Kleine, unabhängige Teams können bottom-up schnelle Entscheidungen treffen und machen die Organisation antifragil
- Das Softwareentwicklungssystem wächst durch Stressoren
- Agile Manifesto: "responding to change over following a plan"
- **A(ntifra)gile** Softwareentwicklung?

Wie antifrägil sind wir?

- DORA Report 2024: 60 Prozent deployen max. einmal pro Woche, 25 Prozent max. einmal pro Monat
- Experimenteller Ansatz mehrheitlich technisch nicht möglich
- Thesen:
 - Auch dort, wo die technischen Voraussetzungen erfüllt sind, sieht die Organisationskultur oft kein antifrägiles Tüfteln vor
 - Betriebswirtschaftliche Konsequenzen in den meisten Teams nicht berücksichtigt — außer "mache nichts, was unerschwinglich teuer ist"

**Wir können noch deutlich
antifragiler werden! 🎉**

Referenzen

- Reinertsen, D.G. (1997). Managing the Design Factory: A Product Developer's Toolkit. Free Press
- Reinertsen, D.G. (2009). The Principles of Product Development Flow: Second Generation Lean Product Development. Celeritas Pub
- Forsgren, N., Humble, J. & Kim, G. (2018). Accelerate: Building and Scaling High Performing Technology Organizations. IT Revolution Press
- Taleb, N. (2012). Antifragile: Things That Gain from Disorder. Random House
- Farley, D. (2021). Modern Software Engineering: Doing What Works to Build Better Software Faster. Addison-Wesley Professional

Danke! Fragen?



Daniel Westheide

daniel.westheide@innoq.com

<https://www.linkedin.com/in/danielwestheide/>

<https://mastodon.social/@kaffeecoder>

innoQ Deutschland GmbH

Krischerstr. 100
40789 Monheim
+49 2173 333660

Ohlauer Str. 43
10999 Berlin

Ludwigstr. 180E
63067 Offenbach

Kreuzstr. 16
80331 München

Hermannstr. 13
20095 Hamburg

Erfststr. 15-17
50672 Köln

Königstorgraben 11
90402 Nürnberg